Public Key Encryption with Equality Test in the Standard Model

Hyung Tae Lee[1], San Ling[1], Jae Hong Seo[2],
Huaxiong Wang[1], and Taek-Young Youn[3]

[1] Division of Mathematical Sciences,
School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore
{hyungtaelee,lingsan,hxwxang}@ntu.edu.sg
[2] Myongji University, Korea
jaehongseo@mju.ac.kr
[3] Electonics and Telecommunications Research Institute, Korea
taekyoung@etri.re.kr

Abstract. Public key encryption with equality test (PKEET) is a cryptosystem that allows a tester who has trapdoors issued by one or more users U_i to perform equality tests on ciphertexts encrypted using public key(s) of U_i. Since this feature has a lot of practical applications including search on encrypted data, several PKEET schemes have been proposed so far. However, to the best of our knowledge, all the existing proposals are proven secure only under the hardness of number-theoretic problems and the random oracle heuristics.

In this paper, we show that this primitive can be achieved not only *generically* from well-established other primitives but also even *without* relying on the random oracle heuristics. More precisely, our generic construction for PKEET employs a two-level hierarchical identity-based encryption scheme, which is selectively secure against chosen plaintext attacks, a strongly unforgeable one-time signature scheme and a cryptographic hash function. Our generic approach toward PKEET has several advantages over all the previous works; it directly leads the first standard model construction and also directly implies the first lattice-based construction. Finally, we show how to extend our approach to the identity-based setting.

Keywords: Public key encryption with equality test, identity-based encryption with equality test, standard model

1 Introduction

Public key encryption with equality test (PKEET), which was first introduced by Yang et al. [23], is a public key encryption (PKE) scheme that allows to perform an equality test on encrypted data using different public keys as well as the same public key. In PKEET schemes, each user issues a trapdoor for equality tests to a tester, and henceforth the tester is able to check the equality between ciphertexts of users who issued the trapdoor to him.[4] This property is of use in various practical applications, such as keyword search on encrypted data, encrypted data partitioning for efficient encrypted data management, personal health record systems, and so on.

As a very similar primitive to PKEET, we may recall public key encryption with keyword search (PKEKS) due to Boneh et al. [6]. In PKEKS schemes, each user issues a token for a keyword to the tester using his/her private key, and henceforth the tester is able to perform an equality test between a message in a ciphertext of the user and the keyword in the issued token. Hence, it can be utilized to check the equality between ciphertexts of the same user. However, the most important feature of PKEET beyond PKEKS is to support equality tests on ciphertexts between different users.

Let us consider more concrete application scenario of PKEET to elaborate advantages of PKEET over PKEKS. In an email service, suppose each user stores his/her emails to the server as encrypted. To support keyword search over encrypted emails efficiently, encrypted keywords are appended to stored emails. The server needs to monitor stored emails to maintain the security of the system. To this end, the server may need to test equality among encrypted keywords. If the system exploits PKEKS for this functionality, the server can perform equality tests on encrypted keywords by using a token for a keyword, issued by the owner of ciphertexts. However, the server needs a different token for each user and each keyword and so it should interact with each user whenever it has a new keyword to be monitored. On the contrary, if the system exploits PKEET, the server can generate a ciphertext of a keyword by himself and perform equality tests.

To support equality tests on encrypted data required for the above scenario, one may consider to exploit other types of encryption schemes, such as deterministic encryption and fully homomorphic encryption, to name a few. However, they are faced with the obstacles that PKEKS has. We will discuss more about relations of PKEET to other primitives including PKEKS in Section 1.2. On the other hand, apart from the above scenario, PKEET can also be utilized as a building block to design advanced cryptographic constructions, e.g., group signatures with controllable linkability [19].

Due to its wide applicability, since Yang et al. proposed the first PKEET realization, various subsequent constructions [20, 22, 17, 16, 11, 15, 14, 12] have been proposed to improve performance or to support advanced functionality. We note that, however, all the existing proposals are proven secure only under the hardness of specific number-theoretic problems and the random oracle heuristics. There is a lot of literature discussing the limitations of the random oracle model (e.g., [8, 10, 13]) and thus, it is highly desirable to achieve each primitive (PKEET in our case) without relying on the random oracle heuristics from both theoretical and practical standpoints.

[4] While Yang et al.'s construction [23] allows anyone to perform equality tests on ciphertexts, all subsequent works consider the existence of the tester.

1.1 Our Contribution

In this paper, our main contribution is a proposal for PKEET schemes. The proposed construction has several advantages over all the previous PKEET constructions. First, our construction does not rely on the random oracle heuristics. Second, our construction does not rely on underlying mathematical structures in the sense that we do not require any number-theoretic assumptions for the security of the proposed construction. More precisely, we employ a 2-level hierarchical identity-based encryption (HIBE) scheme, which is selectively secure against the chosen-plaintext attacks, a strongly unforgeable one-time signature scheme and a cryptographic hash function. Finally, our construction can be easily extended to the identity-based setting since it already employs HIBE as a building block. As a result, we also obtain the first identity-based encryption with equality test (IBEET) in the standard model.

As direct applications of our generic construction, we can obtain PKEET constructions that are proven secure under various types of assumptions, such as pairing-based, lattice-based, and RSA-based ones. In particular, for example, we may obtain a PKEET construction from lattices by exploiting a strongly unforgeable lattice-based signature scheme (e.g., [18]) and a 2-level lattice-based HIBE scheme (e.g., [2, 3]), which satisfies the indistinguishability under selective identity, chosen plaintext attacks (IND-sID-CPA). To the best of our knowledge, it is the first PKEET scheme whose underlying assumptions are solely lattice-based.

There exists a very recent work by Lee et al. [12] that has a similar goal to ours. We clearly note that Lee et al.'s approach is not completely generic in the sense that it essentially uses the computation Diffie-Hellman (CDH) assumption. Furthermore, Lee et al. proved the security of their scheme only in the random oracle model.

Overview of Our Generic Construction. Let us explain a high level intuition for our generic PKEET construction. We begin with a trivial insecure construction and then modify it subsequently. Suppose that each user has a 2-level HIBE scheme $\mathcal{HIBE} = (\mathcal{S}etup, \mathcal{K}eyExt, \mathcal{E}nc, \mathcal{D}ec)$. To generate a ciphertext of a message M, we design an encryption algorithm that performs the following steps. It first selects a random string str from a set $\{0,1\}^*$ as a second level identity and generates two ciphertexts

$$C_0 = \mathcal{E}nc(\mathsf{mpk}, [0.\mathsf{str}], M) \text{ and } C_1 = \mathcal{E}nc(\mathsf{mpk}, [1.\mathsf{str}], H(M))$$

where mpk is a master public key of $\mathcal{HIBE}$, H is a hash function, and $[\mathrm{ID}_1.\mathrm{ID}_2]$ denotes an identity whose i-th level identity is ID_i for $i = 1, 2$. This process outputs (str, C_0, C_1) as a ciphertext of the message M. The decryption algorithm is straightforward; it first generates the secret keys related to str, decrypts C_0 and C_1 by using generated secret keys, and outputs $\mathcal{D}ec(C_0)$ if $H(\mathcal{D}ec(C_0)) = \mathcal{D}ec(C_1)$.

Once a trapdoor algorithm issues a secret key of the identity 1 to the tester, he can generate a secret key of the identity $[1.\mathsf{str}]$ by himself. Using this, he may perform an equality test by decrypting C_1 parts of ciphertexts and then comparing $H(M)$ values. On the other hand, the user who has the master secret key of $\mathcal{HIBE}$ as his/her secret key, can generate a secret key of the identity $[0.\mathsf{str}]$ and obtain the correct message by using it to decrypt C_0. However, the above construction does not satisfy the security against adaptive chosen ciphertext attacks (CCA2). Although C_0 and C_1 are linked by str and there is a process to check the link between C_0 and C_1 during the execution of the decryption algorithm, it is not enough to prevent chosen ciphertext attacks. In fact, the adversary can obtain the message in C_0 by requesting a query on a modified ciphertext $(\mathsf{str}, C_0, \cdot)$ to the decryption oracle in the security game. To prevent such a chosen ciphertext attack, almost all previous PKEET constructions in the random oracle model (including the semi-generic one [12]) have embedded an instance of the CDH problem.

To achieve the CCA2 security in a generic way, we borrow an idea of the Canetti, Halevi, and Katz (CHK) transformation to obtain a CCA2 secure PKE scheme from an IND-sID-CPA secure identity-based encryption (IBE) scheme using a strongly unforgeable one-time signature scheme in the standard model. We first assign a signature scheme to each user. By adapting their technique, we modify our encryption algorithm so that it first generates a pair of a signing key and a verification key $(\mathsf{sk_s}, \mathsf{vk_s})$ and uses the verification key $\mathsf{vk_s}$ as the second level identity, instead of a random string str. After generating two ciphertexts,

$$C_0 = \mathcal{E}nc(\mathsf{mpk}, [0.\mathsf{vk_s}], M) \text{ and } C_1 = \mathcal{E}nc(\mathsf{mpk}, [1.\mathsf{vk_s}], H(M)),$$

it additionally generates a signature σ on the ciphertexts C_0 and C_1 using a signing key $\mathsf{sk_s}$. Thereafter, it outputs

$$(\mathsf{vk_s}, C_0, C_1, \sigma)$$

as a ciphertext.

Informally, we can regard that an adversary has to succeed in forging a signature on some message with respect to the verification key in the challenge ciphertext, in order to receive the correct response from a decryption oracle query on a ciphertext obtained by modifying the challenge ciphertext. We show that our construction achieves one-wayness under adaptive chosen ciphertext attacks (OW-CCA2) against Type-I adversaries, who have a trapdoor for equality tests, and the indistinguishability under adaptive chosen ciphertext attacks (IND-CCA2) against Type-II adversaries, who do not have a trapdoor, if the exploited HIBE scheme is IND-sID-CPA secure and the exploited signature scheme is strongly unforgeable in the standard model. As a result, we obtain the first PKEET construction in the standard model.

Finally, we discuss an extension of our construction to the identity-based setting by replacing a 2-level HIBE scheme with a 3-level one. In our modification, the first level of the employed 3-level HIBE scheme is reserved for an identity, while its second and third levels play the same roles as the first and second levels of our PKEET construction, respectively. The derived outcome is the first IBEET construction in the standard model as well.

1.2 Related Works

In this subsection, we look at some primitives related to PKEET. We also introduce existing works related to the technique used in our construction.

PKE/IBE with Equality Test. Yang et al. [23] first introduced the concept of PKEET, which allows anyone to check whether two ciphertexts contain the same message or not, under different public keys as well as the same public key. Following their work, Tang [22] proposed a variant, so called all-or-nothing PKEET, which allows only testers authorized by two users to perform equality tests on all ciphertexts of those users. Subsequently, there have been proposed various PKEET constructions [20, 21, 17, 16, 11, 14, 12] to improve performance or to support advanced functionality.

In the identity-based setting, Ma [15] proposed a system model for IBEET, which is an identity-based version of Tang's all-or-nothing PKEET model, and presented an instantiation of IBEET. Later, Lee et al. [12] showed that their semi-generic construction for PKEET can be extended to the identity-based setting and the derived outcome achieves better security than Ma's scheme.

On the other hand, all existing PKEET/IBEET schemes are constructed in the random oracle model. Moreover, all previous proposals rely on specific number-theoretic assumptions.

PKE/IBE with Keyword Search. PKE with keyword search (PKEKS) is a PKE scheme that supports the functionality to perform an equality test between a keyword embedded in a tag and a message in a ciphertext [6]. It is quite similar to PKEET in the sense that both are able to check the equality on encrypted data. However, the main difference between the two schemes is that PKEKS allows tests on ciphertexts under only a fixed public key related to the issued tag, whereas PKEET allows equality tests on ciphertexts under different public keys as well as the same public key.

Abdalla et al. [1] considered an extension of PKEKS to the identity-based setting, which has similar features to those of IBEET. As similar with the relation between PKEKS and PKEET, IBE with keyword search (IBEKS) also allows to perform equality tests on ciphertexts under a fixed identity related to the issued tag, whereas IBEET allows equality tests on ciphertexts under different identities as well as the same identity.

Deterministic Encryption. The notion of deterministic encryption was first initiated by Bellare et al. [4]. For testers, PKEET has the very similar feature to deterministic encryption in the sense that it supports equality tests on ciphertexts, although PKEET is a probabilistic encryption. Hence, PKEET can be exploited to many applications of deterministic encryption. On the other hand, in the view of others who do not have a trapdoor, PKEET works just like traditional PKE schemes and so it is expected that PKEET achieves better security than deterministic encryption. Further, as the same as PKEKS, deterministic encryption supports equality tests between ciphertexts of the same user only.

Transformation from (H)IBE to Other Primitives. In our construction, we borrow the strategy of the CHK transformation that obtains a CCA2 secure

PKE scheme from an IND-sID-CPA secure IBE scheme with exploiting strongly unforgeable one-time signatures. We note that there is a generic transformation to obtain a PKEKS scheme using an IBE scheme. In [1], Abdalla et al. provided a transformation of an anonymous IBE scheme to a secure PKEKS scheme. Further, as similar to ours, they also extended their transformation to the identity-based setting by replacing the underlying IBE scheme with an HIBE one. However, its design principle is quite different from ours: For example, while our construction uses a verification key of the underlying signature scheme as an identity and encrypts a message using it, their transformation generates a random message and encrypts it with a keyword in PKEKS as an identity.

1.3 Organization of the Paper

Section 2 presents basic definitions of several components exploited in our constructions. In Section 3, we introduce syntax and the security model of PKEET. Section 4 provides our PKEET scheme and Section 5 analyzes its security. We discuss about extensions of our construction in Section 6 and give concluding remarks in Section 7. The details about formal definitions of IBEET and our IBEET construction are given in Appendices.

2 Preliminaries

In this section, we look at basic definitions of components that will be exploited to design our PKEET schemes.

Notation. Throughout the paper, λ denotes a security parameter. For an algorithm A, $A \rightarrow a$ and $A \not\rightarrow a$ denote that A outputs a and A does not, respectively. A function $\nu : \mathbb{N} \rightarrow \mathbb{R}$ is negligible in λ if for all positive polynomials $p(\cdot)$ and sufficiently large λ, $\nu(\lambda) \leq \frac{1}{p(\lambda)}$. $\mathsf{negl}(\lambda)$ represents a negligible function in λ.

Hierarchical Identity-Based Encryption. We begin with presenting the definition of HIBE and its security notion, which will be utilized as a building block of our generic construction for PKEET schemes.

Definition 1 (Hierarchical Identity-Based Encryption). *A hierarchical identity-based encryption (HIBE) scheme consists of the following four probabilistic polynomial-time (PPT) algorithms:*

- $\mathcal{S}etup(\lambda)$*: On input a security parameter λ, it outputs a pair of a master public key and a master secret key* $(\mathsf{mpk}, \mathsf{msk})$*. It is assumed that* mpk *contains the message space $\mathcal{M}$ and the identity space $\mathcal{I}$.*
- $\mathcal{K}eyExt(\mathsf{sk}_{\mathrm{ID}}, \mathrm{ID}')$*: It takes a secret key $\mathsf{sk}_{\mathrm{ID}}$ of an identity* ID *and* ID*'s descendant* ID' *(that is,* ID *is a prefix of* ID'*) as inputs and outputs a secret key $\mathsf{sk}_{\mathrm{ID}'}$ of the identity ID'. If ID' is a first-level user, then $\mathsf{sk}_{\mathrm{ID}}$ should be the master secret key* msk*.*

– $\mathcal{E}nc(\mathsf{mpk}, \mathrm{ID}, M)$*: It takes* mpk*, a recipient's identity* ID*, and a message* $M \in \mathcal{M}$ *as inputs and outputs a ciphertext* ct.

– $\mathcal{D}ec(\mathsf{sk}_{\mathrm{ID}}, \mathsf{ct})$*: It takes a secret key* $\mathsf{sk}_{\mathrm{ID}}$ *of an identity* ID *and a ciphertext* ct *as inputs and outputs a message* M'.

We say an HIBE scheme is *correct* if for any output $(\mathsf{mpk}, \mathsf{msk})$ of $\mathcal{S}etup(\lambda)$, message $M \in \mathcal{M}$, identity $\mathrm{ID}' \in \mathcal{I}$, and output $\mathsf{sk}_{\mathrm{ID}'}$ of $\mathcal{K}eyExt(\mathsf{sk}_{\mathrm{ID}}, \mathrm{ID}')$ with any ID''s ancestor ID, it holds

$$\Pr[\mathcal{D}ec(\mathsf{sk}_{\mathrm{ID}'}, \mathcal{E}nc(\mathsf{mpk}, \mathrm{ID}', M)) \to M] = 1,$$

where the probability goes over all randomness values used in all corresponding algorithms. Furthermore, we require that for any identity ID, its delegated secret keys obtained from its any ancestor have the same distribution. We remark that an IBE scheme is an HIBE scheme where all identities are at level 1.

For our construction, we need a selective identity secure HIBE scheme against chosen plaintext attacks.

Definition 2 (IND-sID-CPA). *An IBE or HIBE scheme* $\mathcal{E} = (\mathcal{S}etup, \mathcal{K}eyExt, \mathcal{E}nc, \mathcal{D}ec)$ *is IND-sID-CPA secure if for any PPT adversary* $\mathcal{A}$*, the advantage of the following game between the adversary* $\mathcal{A}$ *and the challenger* $\mathcal{C}$ *is negligible in the security parameter* λ*:*

1. **Init:** $\mathcal{A}$ *outputs a target identity* ID^*.
2. **Setup:** $\mathcal{C}$ *runs* $\mathcal{S}etup(\lambda) \to (\mathsf{mpk}, \mathsf{msk})$*, forwards the master public key* mpk *to* $\mathcal{A}$*, and keeps the master secret key* msk *private.*
3. **Phase 1:** $\mathcal{A}$ *may issue private key queries on* ID_i*, which is neither* ID^* *nor an ancestor of* ID^**, polynomially many times. To respond to queries,* $\mathcal{C}$ *runs the* $\mathcal{K}eyExt$ *algorithm to generate a secret key* $\mathsf{sk}_{\mathrm{ID}_i}$ *of the requested identity* ID_i *and returns* $\mathsf{sk}_{\mathrm{ID}_i}$ *to* $\mathcal{A}$.
4. **Challenge:** $\mathcal{A}$ *outputs two messages* M_0, M_1 *of the same length and forwards* $\mathcal{C}$ *them.* $\mathcal{C}$ *selects a random bit* $b \in \{0, 1\}$*, runs* $\mathcal{E}nc(\mathsf{mpk}, \mathrm{ID}^*, M_b) \to \mathsf{ct}_b^*$*, and passes* ct_b^* *as the challenge ciphertext to* $\mathcal{A}$.
5. **Phase 2:** $\mathcal{A}$ *may issue private key queries on* ID_i*, which is neither* ID^* *nor an ancestor of* ID^**, polynomially many times.* $\mathcal{C}$ *responds as in Phase 1.*
6. **Guess:** $\mathcal{A}$ *outputs* $b' \in \{0, 1\}$.

We say $\mathcal{A}$ *wins if* $b = b'$ *and the advantage of* $\mathcal{A}$ *in the above game is defined to*

$$\mathbf{Adv}_{\mathcal{A},\mathcal{E}}^{\text{IND-sID-CPA}}(\lambda) := \left| \Pr[b = b'] - \frac{1}{2} \right|.$$

We remark that in the above game, if $\mathcal{A}$ sends a target identity ID^* together with two messages M_0, M_1 at the challenge step, not at the beginning of the game, then we say that a scheme $\mathcal{E}$ is IND-ID-CPA secure.

Signatures. To achieve the CCA2 security of our PKEET construction, we will borrow the strategy of the CHK transformation [9] that was originally proposed to convert IND-sID-CPA secure IBE schemes into CCA2 secure PKE schemes with employing a strongly unforgeable one-time signature scheme in the standard model. Here, we describe the formal definition of signature schemes and provide its security notion required for our construction.

Definition 3 (Digital Signature Scheme). *A digital signature scheme consists of the following three PPT algorithms:*

- $\mathsf{G}(\lambda)$: *It takes a security parameter λ as an input and outputs a pair of a verification key and a signing key $(\mathsf{vk_s}, \mathsf{sk_s})$.*
- $\mathsf{S}(\mathsf{sk_s}, M)$: *On input the singing key $\mathsf{sk_s}$ and a message M, it outputs a signature σ.*
- $\mathsf{V}(\mathsf{vk_s}, M, \sigma)$: *It takes the verification key $\mathsf{vk_s}$, a message M, and a signature σ as inputs, and returns 1 or 0.*

We say that a digital signature scheme is *correct* if for any output $(\mathsf{vk_s}, \mathsf{sk_s})$ of $\mathsf{G}(\lambda)$ and any message M, it holds

$$\Pr[\mathsf{V}(\mathsf{vk_s}, M, \mathsf{S}(\mathsf{sk_s}, M)) \rightarrow 1] = 1,$$

where the probability goes over all randomness values used in all corresponding algorithms. We also say that a digital signature is *one-time* if for each verification key the signing algorithm runs only once; that is, in the following security model, we may assume that the adversary attacking a one-time signature scheme can issue a signing query only once.

Definition 4 (Strong Unforgeability). *A signature scheme $\mathsf{Sig} = (\mathsf{G}, \mathsf{S}, \mathsf{V})$ is strongly unforgeable under an adaptive chosen message attack if for any PPT adversary $\mathcal{A}$, the probability that the adversary wins in the following game with the challenger $\mathcal{C}$ is negligible in the security parameter λ:*

1. **Setup:** *$\mathcal{C}$ runs $\mathsf{G}(\lambda) \rightarrow (\mathsf{vk_s}, \mathsf{sk_s})$, passes the verification key $\mathsf{vk_s}$ to $\mathcal{A}$, and keeps the signing key $\mathsf{sk_s}$ private.*
2. **Signature Queries:** *$\mathcal{A}$ issues signing queries on messages M_i polynomially many times. For each M_i, $\mathcal{C}$ runs $\mathsf{S}(\mathsf{sk_s}, M_i) \rightarrow \sigma_i$ and sends σ_i to $\mathcal{A}$. These queries may be requested adaptively so that each query may depend on the replies to the previous queries.*
3. **Output:** *$\mathcal{A}$ outputs a pair (M, σ).*

The adversary wins if $\mathsf{V}(\mathsf{vk_s}, M, \sigma) \rightarrow 1$ and (M, σ) is not generated during the signature query phase.

Properties of Hash Functions. To show the correctness and security of our construction, we will exploit the following properties of hash functions.

Definition 5 (One-way Functions). *A function H is one-way if the following two conditions hold:*

1. *There exists a polynomial-time algorithm to compute H.*

2. *For any PPT algorithm A, it holds that*

$$\Pr[A(\lambda, H, y) \to x \textit{ such that } H(x) = y] \leq \mathsf{negl}(\lambda)$$

where $y = H(x')$ for a randomly chosen x' from the domain.

Definition 6 (Collision Resistant Hash Functions). *A family of hash functions $\{H_s\}$ is collision resistant if the following three conditions hold:*

1. *There exists a PPT algorithm* $\mathsf{Gen}(\lambda)$ *that outputs an index s.*
2. *There exists a polynomial-time algorithm to compute H_s.*
3. *For any PPT algorithm A, it holds that*

$$\Pr[\mathsf{Gen}(\lambda) \to s, A(s) \to (x, x') \textit{ such that } x \neq x' \textit{ and } H_s(x) = H_s(x')] \leq \mathsf{negl}(\lambda).$$

3 Syntax and Security Model

In this section, we review the concept of PKEET and its security model. There have been several types of definitions of PKEET with slight differences. Among them, our system model follows the concept of all-or-nothing PKEET scheme, proposed by Tang [22].

3.1 Syntax

System Model for Our PKEET. Our PKEET system consists of users (e.g., senders and receivers) and testers (e.g., the server): In the system, a sender encrypts a data using a receiver's public key and sends a ciphertext to the receiver. The receiver may decrypt his/her ciphertexts using his/her secret key and/or store ciphertexts at the server. Once the receiver wants to delegate the test capability for all of his/her ciphertexts, he/she issues a trapdoor for equality tests to a tester who can access to the server that stores encrypted data. Hereafter, the tester is able to perform equality tests on ciphertexts under the public key of the receiver who delegated the test authority for his/her ciphertexts to the tester.

Definitions of PKEET. Before giving a definition of PKEET, we first note that PKEET is a multi-user setting and we assume that each user is assigned an index i for $1 \leq i \leq N$, where N is the number of users in the system. We use U_i to denote the i-th user. Furthermore, for notational convenience, we use a subscripted index to indicate keys and ciphertexts for each user, e.g., pk_i is a public key for user U_i.

Definition 7 (Public Key Encryption with Equality Test). *A public key encryption with equality test (PKEET) consists of six polynomial-time algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Td}, \mathsf{Test})$*, specified as follows.*

- $\mathsf{Setup}(\lambda)$*: On input a security parameter* λ*, it outputs a system parameter* params*, which includes the message space* $\mathcal{M}$*. It is assumed that all other algorithms take* params *as an input implicitly, though it is not stated.*
- $\mathsf{KeyGen}(\mathsf{params})$*: It takes* params *as an input and outputs a pair of user's public and secret keys* $(\mathsf{pk}, \mathsf{sk})$*.*
- $\mathsf{Enc}(\mathsf{pk}, M)$*: It takes* pk *and a message* $M \in \mathcal{M}$ *as inputs, and outputs a ciphertext* ct*.*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$*: It takes* sk *and a ciphertext* ct *as inputs, and outputs a message* M' *or* $\perp$*.*
- $\mathsf{Td}(\mathsf{sk}_i)$*: On input a secret key* sk_i *of a user* U_i*, it outputs a trapdoor* td_i*.*
- $\mathsf{Test}(\mathsf{td}_i, \mathsf{ct}_i, \mathsf{td}_j, \mathsf{ct}_j)$*: It takes two ciphertexts* ct_i *and* ct_j *and two trapdoors* td_i *and* td_j *as inputs, and outputs* 0 *or* 1*.*

Correctness. Basically, PKEET is a PKE scheme, so that we require that for any $\mathsf{Setup}(\lambda) \to \mathsf{params}$ and $\mathsf{KeyGen}(\mathsf{params}) \to (\mathsf{pk}_i, \mathsf{sk}_i)$ and for any message $M \in \mathcal{M}$ and index i, $\mathsf{Dec}(\mathsf{sk}_i, \mathsf{Enc}(\mathsf{pk}_i, M)) = M$ always holds. For the functionality of Td and Test algorithms, we require the following two additional conditions to be satisfied. For any $\mathsf{Setup}(\lambda) \to \mathsf{params}$, $\mathsf{KeyGen}(\mathsf{params}) \to (\mathsf{pk}_i, \mathsf{sk}_i)$, $\mathsf{KeyGen}(\mathsf{params}) \to (\mathsf{pk}_j, \mathsf{sk}_j)$, $\mathsf{Td}(\mathsf{sk}_i) \to \mathsf{td}_i$, $\mathsf{Td}(\mathsf{sk}_j) \to \mathsf{td}_j$, $\mathsf{Enc}(\mathsf{pk}_i, M_i) \to \mathsf{ct}_i$, and $\mathsf{Enc}(\mathsf{pk}_j, M_j) \to \mathsf{ct}_j$ with $M_i, M_j \in \mathcal{M}$,

1. $\Pr\left[\mathsf{Test}(\mathsf{td}_i, \mathsf{ct}_i, \mathsf{td}_j, \mathsf{ct}_j) \to 1\right] = 1$ if $M_i = M_j$, regardless of whether $i = j$ or $i \neq j$;
2. $\Pr[\mathsf{Test}(\mathsf{td}_i, \mathsf{ct}_i, \mathsf{td}_j, \mathsf{ct}_j) \to 1]$ is negligible in the security parameter λ for any ciphertexts ct'_i and ct'_j such that $\mathsf{Dec}(\mathsf{sk}_i, \mathsf{ct}'_i) \neq \mathsf{Dec}(\mathsf{sk}_j, \mathsf{ct}'_j)$.

3.2 Security Model

For the security of PKEET, we consider two different scenarios according to whether the adversary has a trapdoor for the target user or not. In case where the adversary has the trapdoor for the equality test, we cannot expect the indistinguishability-based security notion for PKEET, and, probably, the onewayness is the best achievable security. Otherwise, we can define the indistinguishability security notion for PKEET.

More precisely, we consider the following two types of adversaries for our PKEET system model.

- Type-I adversary: This type of adversaries has the trapdoor for the target user's ciphertexts and so the adversary can perform an equality test with the challenge ciphertext. Hence, we consider that the aim of the adversary is to reveal the message contained in the challenge ciphertext.
- Type-II adversary: This type of adversaries has no trapdoor for the target user's ciphertexts and so the adversary cannot perform an equality test with the challenge ciphertext. Hence, we consider that the aim of the adversary is to distinguish whether the challenge ciphertext contains which message between two candidates.

We first provide the formal security definition for PKEET constructions against Type-I adversaries below.

Definition 8 (OW-CCA2 against Type-I Adversaries). *A PKEET scheme is OW-CCA2 secure against Type-I adversaries if for any PPT adversary $\mathcal{A}$, the success probability of $\mathcal{A}$ in the following game with the challenger $\mathcal{C}$ is negligible in the security parameter λ: Let U_t be the target user.*

1. **Setup:** *$\mathcal{C}$ runs $\mathsf{Setup}(\lambda) \to \mathsf{params}$ and sends the system parameter params to $\mathcal{A}$. Then, $\mathcal{C}$ runs $\mathsf{KeyGen}(\mathsf{params}) \to (\mathsf{pk}_i, \mathsf{sk}_i)$ for $1 \leq i \leq N$ and passes all pk_i's to $\mathcal{A}$.*
2. **Phase 1:** *$\mathcal{A}$ may query the following oracles polynomially many times adaptively and in any order. The constraint is that an index t cannot be queried to the key extraction oracle $\mathcal{O}^{\mathsf{sk}}$.*
   - *$\mathcal{O}^{\mathsf{sk}}$: an oracle that on input an index i, returns the U_i's secret key sk_i.*
   - *$\mathcal{O}^{\mathsf{Dec}}$: an oracle that on input a pair of an index i and a ciphertext ct_i, returns the output of $\mathsf{Dec}(\mathsf{sk}_i, \mathsf{ct}_i)$ using the U_i's secret key sk_i.*
   - *$\mathcal{O}^{\mathsf{Td}}$: an oracle that on input an index i, returns td_i by running $\mathsf{Td}(\mathsf{sk}_i) \to \mathsf{td}_i$ with the U_i's secret key sk_i.*
3. **Challenge:** *$\mathcal{C}$ chooses a random message M from the message space $\mathcal{M}$, runs $\mathsf{Enc}(\mathsf{pk}_t, M) \to \mathsf{ct}_t^*$, and sends ct_t^* to $\mathcal{A}$.*
4. **Phase 2:** *For $\mathcal{A}$'s queries, $\mathcal{C}$ responds as in Phase 1. The constraints for $\mathcal{A}$'s queries are that*
   (a) *the index t cannot be queried to the key extraction oracle $\mathcal{O}^{\mathsf{sk}}$;*
   (b) *the pair of the index t and the ciphertext ct_t^* cannot be queried to the decryption oracle $\mathcal{O}^{\mathsf{Dec}}$.*
5. **Guess:** *$\mathcal{A}$ outputs M'.*

The adversary $\mathcal{A}$ wins in the above game if $M = M'$ and the success probability of $\mathcal{A}$ is defined to

$$\mathbf{Adv}^{\text{OW-CCA2}}_{\mathcal{A},\text{PKEET}}(\lambda) := \Pr[M = M'].$$

Remark 1. (Constraint on the message space) If the size of the message space is polynomial in the security parameter or the min-entropy of the message distribution is much lower than the security parameter, a Type-I adversary who has a trapdoor for the challenge ciphertext, can reveal the message in the challenge ciphertext in polynomial time or sufficiently small exponential time in the security parameter, by performing equality tests with the challenge ciphertext and other ciphertexts of all messages, generated by himself. To prevent these trivial attacks, it is assumed that the size of the message space is exponential in the security parameter and the min-entropy of the message distribution is sufficiently higher than the security parameter.

Now, we present the formal security definition for PKEET constructions against Type-II adversaries.

Definition 9 (IND-CCA2 against Type-II Adversaries). *A PKEET scheme is IND-CCA2 secure against Type-II adversaries if for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game with the challenger $\mathcal{C}$ is negligible in the security parameter λ: Let U_t be the target user.*

1. **Setup:** *This step is the same as that of the OW-CCA2 security game in Definition 8.*
2. **Phase 1:** *This step is almost the same as that of the OW-CCA2 security game in Definition 8, except that the constraint is that an index t cannot be queried to the trapdoor extraction oracle $\mathcal{O}^{\mathsf{Td}}$ as well as the key extraction oracle $\mathcal{O}^{\mathsf{sk}}$.*
3. **Challenge:** *$\mathcal{A}$ chooses two message $M_0, M_1 \in \mathcal{M}$ of the same length and passes $\mathcal{C}$ them. $\mathcal{C}$ selects a random bit $b \in \{0,1\}$, runs $\mathsf{Enc}(\mathsf{pk}_t, M_b) \to \mathsf{ct}^*_{t,b}$, and sends $\mathsf{ct}^*_{t,b}$ to $\mathcal{A}$.*
4. **Phase 2:** *For $\mathcal{A}$'s queries, $\mathcal{C}$ responds as in Phase 1. The constraints for $\mathcal{A}$'s queries are that*
   (a) *the index t cannot be queried to the key extraction oracle $\mathcal{O}^{\mathsf{sk}}$ and the trapdoor extraction oracle $\mathcal{O}^{\mathsf{Td}}$;*
   (b) *the pair of the index t and the ciphertext $\mathsf{ct}^*_{t,b}$ cannot be queried to the decryption oracle $\mathcal{O}^{\mathsf{Dec}}$.*
5. **Guess:** *$\mathcal{A}$ outputs b'.*

The adversary $\mathcal{A}$ wins in the above game if $b = b'$ and the advantage of $\mathcal{A}$ is defined to

$$\mathbf{Adv}^{\text{IND-CCA2}}_{\mathcal{A},\text{PKEET}}(\lambda) := \left| \Pr[b = b'] - \frac{1}{2} \right|.$$

3.3 Extension to Identity-Based Settings

One can easily extend the definition of PKEET to the identity-based setting. Let us briefly point out differences only. (Refer to Appendix A for formal definitions of IBEET.) The setup algorithm Setup outputs a pair of a master public key mpk and a master secret key msk, instead of a system parameter params. The key generation algorithm KeyGen takes mpk and an identity ID, and outputs a secret key of the identity ID, sk_{ID}. All other algorithms are almost the same as the public key setting, except a slight adjustment for an identity instead of a public key.

In the security model, there is a small but important difference. In the PKEET setting, the number of all possible users are N and all corresponding public keys should be sent to the adversary. Here, we implicitly assume that N is polynomially bounded in λ. However, in the identity-based setting, we do not restrict which identities will be joined in the system among exponentially many candidates. That is, the adversary can decide who will join in by asking a secret key of an identity to the corresponding oracle during the security game. The rest part in the security model remains almost unchanged, except a fine-tuning for an identity instead of a public key. We note that such a small difference between the numbers of manipulable users in systems should be carefully handled in the security proof. Hence, in the security proof for PKEET, one may assume that the simulator knows the target public key pk_t in advance at the beginning of the simulation, at the price of polynomial reduction loss. However, in the security proof for IBEET, if one would like to make a construction that achieves the adaptive ID security, not the selective ID security, one may not be able to assume that the simulator knows the target identity at the very beginning of the simulation because there are exponentially many candidates. It causes a situation that our IBEET construction should employ an IND-ID-CPA secure HIBE scheme, whereas our PKEET construction employs an IND-sID-CPA secure one.

4 Our PKEET Construction

In this section, we present our construction for PKEET by exploiting traditional 2-level HIBE schemes and signature schemes. Hereafter, $[\mathrm{ID}_1.\mathrm{ID}_2]$ denotes an identity of a 2-level HIBE scheme $\mathcal{HIBE} = (\mathcal{S}etup, \mathcal{K}eyExt, \mathcal{E}nc, \mathcal{D}ec)$ where ID_1 is the first level identity and ID_2 is the second level identity. $[M_0 \| M_1]$ denotes the concatenation of messages M_0 and M_1.

Our Strategy. Our design strategy is based on the transformation from an IND-sID-CPA secure IBE scheme into an IND-CCA2 secure PKE scheme in the standard model, proposed by Canetti, Halevi, and Katz (CHK) [9], which has a ciphertext of the form $\mathsf{ct} = (\mathsf{vk_s}, C, \sigma)$, where $\mathsf{G}(\lambda) \rightarrow (\mathsf{vk_s}, \mathsf{sk_s})$, $\mathcal{E}nc(\mathsf{mpk}, \mathsf{vk_s}, M) \rightarrow C$, and $\mathsf{S}(\mathsf{sk_s}, C) \rightarrow \sigma$ for a signature scheme $\mathsf{Sig} = (\mathsf{G}, \mathsf{S}, \mathsf{V})$ and an IBE scheme $\mathcal{IBE} = (\mathcal{S}etup, \mathcal{K}eyExt, \mathcal{E}nc, \mathcal{D}ec)$. More precisely, in the encryption algorithm constructed by the CHK transformation, it performs as follows: 1) generate a pair of a signing key and a verification key $(\mathsf{vk_s}, \mathsf{sk_s})$, 2) encrypt a message M using the encryption algorithm $\mathcal{E}nc$ of the underlying IBE scheme $\mathcal{IBE}$ with a verification key $\mathsf{vk_s}$ as an identity, and finally 3) sign the obtained IBE ciphertext C using the signing key $\mathsf{sk_s}$. Then, the resulting scheme is an IND-CCA2 secure PKE scheme if the underlying signature scheme Sig is strongly unforgeable and the underlying IBE scheme $\mathcal{IBE}$ is IND-sID-CPA secure.

In our construction, to support an equality test, we apply the CHK transformation to two ciphertexts of a message and its hash value at once using a 2-level HIBE scheme instead of an IBE scheme. For given a signature scheme $\mathsf{Sig} = (\mathsf{G}, \mathsf{S}, \mathsf{V})$ and a 2-level HIBE scheme $\mathcal{HIBE} = (\mathcal{S}etup, \mathcal{K}eyExt, \mathcal{E}nc, \mathcal{D}ec)$, the proposed encryption process for a message M is as follows: 1) $\mathsf{G}(\lambda) \rightarrow (\mathsf{vk_s}, \mathsf{sk_s})$, 2) $\mathcal{E}nc(\mathsf{mpk}, [0.\mathsf{vk_s}], M) \rightarrow C_0$, 3) $\mathcal{E}nc(\mathsf{mpk}, [1.\mathsf{vk_s}], H(M)) \rightarrow C_1$ for a hash function H, 4) $\mathsf{S}(\mathsf{sk_s}, [C_0 \| C_1]) \rightarrow \sigma$, and 5) output $\mathsf{ct} = (\mathsf{vk_s}, C_0, C_1, \sigma)$.

A ciphertext of the above construction includes two ciphertexts C_0, C_1 of the underlying HIBE scheme. The former is an encryption of the message, so it enables to obtain the exact message by decrypting it using a decryption key for the identity $[0.\mathsf{vk_s}]$. On the other hand, the latter is an encryption of a hash value of the message, so it enables to perform an equality test by decrypting it using a decryption key for the identity $[1.\mathsf{vk_s}]$ and comparing with them of other ciphertexts.

Informally, when the underlying signature scheme is strongly unforgeable and the underlying HIBE scheme is IND-sID-CPA secure, if the adversary does not have a decryption key for the identity $[1.\mathsf{vk_s}^*]$ that is used in the challenge ciphertext, then the proposed scheme still remains to be IND-CCA2 secure as constructions obtained by the CHK transformation. Otherwise, all the situation is the same as the previous, except that the adversary additionally knows the hash value of the message, $H(M)$. Therefore, it cannot achieve the indistinguishability between messages, but we expect the one-wayness of the message information if H is a one-way function.

Description. We provide a full description of our PKEET construction below.

- $\mathsf{Setup}(\lambda)$: Given a security parameter λ, generate
  1. a 2-level HIBE scheme $\mathcal{HIBE} = (\mathcal{S}etup, \mathcal{K}eyExt, \mathcal{E}nc, \mathcal{D}ec)$,
  2. a hash function $H : \{0,1\}^* \to \mathcal{M}$ for the message space $\mathcal{M}$ of $\mathcal{HIBE}$, and
  3. a digital signature scheme $\mathsf{Sig} = (\mathsf{G}, \mathsf{S}, \mathsf{V})$.

  It outputs a system parameter $\mathsf{params} = \{H, \mathcal{HIBE}, \mathsf{Sig}\}$. We implicitly set the message space of our PKEET scheme to the message space $\mathcal{M}$ of $\mathcal{HIBE}$.
- $\mathsf{KeyGen}(\mathsf{params})$: On input params, it runs $\mathcal{S}etup(\lambda) \to (\mathsf{mpk}, \mathsf{msk})$ and outputs a public key $\mathsf{pk} = \mathsf{mpk}$ and a secret key $\mathsf{sk} = \mathsf{msk}$.
- $\mathsf{Enc}(\mathsf{pk}, M)$: It takes pk and a message $M \in \mathcal{M}$ as inputs and runs
  1. $\mathsf{G}(\lambda) \to (\mathsf{vk_s}, \mathsf{sk_s})$,
  2. $\mathcal{E}nc(\mathsf{pk}, [0.\mathsf{vk_s}], M) \to C_0$,
  3. $\mathcal{E}nc(\mathsf{pk}, [1.\mathsf{vk_s}], H(M)) \to C_1$, and
  4. $\mathsf{S}(\mathsf{sk_s}, [C_0 \| C_1]) \to \sigma$.

  It outputs a ciphertext $\mathsf{ct} = (\mathsf{vk_s}, C_0, C_1, \sigma)$.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: On input ct, parse ct to $(\mathsf{vk_s}, C_0, C_1, \sigma)$. Then, it performs as follows:
  1. Run $\mathcal{K}eyExt(\mathsf{sk}, [i.\mathsf{vk_s}]) \to \mathsf{sk}_{[i.\mathsf{vk_s}]}$ for $i = 0, 1$.
  2. Decrypt C_0 and C_1 by running $\mathcal{D}ec(\mathsf{sk}_{[0.\mathsf{vk_s}]}, C_0) \to M'$ and $\mathcal{D}ec(\mathsf{sk}_{[1.\mathsf{vk_s}]}, C_1) \to h'$.
  3. If $h' = H(M')$ and $\mathsf{V}(\mathsf{vk_s}, [C_0 \| C_1], \sigma) \to 1$, then output M'. Otherwise, output $\perp$.
- $\mathsf{Td}(\mathsf{sk}_i)$: It takes a user U_i's secret key sk_i as an input, runs $\mathcal{K}eyExt(\mathsf{sk}_i, 1) \to \mathsf{sk}_{i,1}$, and outputs $\mathsf{td}_i = \mathsf{sk}_{i,1}$.

– $\mathsf{Test}(\mathsf{td}_i, \mathsf{ct}_i, \mathsf{td}_j, \mathsf{ct}_j)$: It takes trapdoors $\mathsf{td}_i, \mathsf{td}_j$ and ciphertexts $\mathsf{ct}_i, \mathsf{ct}_j$ for users U_i, U_j, respectively, as inputs. For $k = i, j$,

1. parse ct_k to $(\mathsf{vk}_{\mathsf{s},k}, C_{k,0}, C_{k,1}, \sigma_k)$,
2. run $\mathcal{K}ey\mathcal{E}xt(\mathsf{td}_k, [1.\mathsf{vk}_{\mathsf{s},k}]) \to \mathsf{sk}_{k,[1.\mathsf{vk}_{\mathsf{s},k}]}$, and
3. run $\mathcal{D}ec(\mathsf{sk}_{k,[1.\mathsf{vk}_{\mathsf{s},k}]}, C_{k,1}) \to h'_k$.

If $h'_i = h'_j$, then output 1. Otherwise, output 0.

Correctness. The following theorem demonstrates the correctness of our PKEET construction.

Theorem 1. *Our PKEET construction is correct if the underlying HIBE scheme $\mathcal{HIBE}$ and signature scheme Sig are correct, and the employed hash function H is collision resistant.*

Proof. Let $\mathsf{ct} = (\mathsf{vk}_\mathsf{s}, C_0, C_1, \sigma)$ be a valid ciphertext of message M obtained by running $\mathsf{Enc}(\mathsf{pk}, M)$ where pk is a public key generated by running $\mathsf{KeyGen}(\mathsf{params})$ and params is an outcome of the Setup algorithm. That is, $\mathsf{G}(\lambda) \to (\mathsf{vk}_\mathsf{s}, \mathsf{sk}_\mathsf{s})$, $\mathcal{E}nc(\mathsf{pk}, [0.\mathsf{vk}_\mathsf{s}], M) \to C_0$, $\mathcal{E}nc(\mathsf{pk}, [1.\mathsf{vk}_\mathsf{s}], H(M)) \to C_1$, and $\mathsf{S}(\mathsf{sk}_\mathsf{s}, [C_0 \| C_1])$ $\to \sigma$. Because $\mathcal{HIBE}$ is correct,

$$\begin{aligned}\mathcal{D}ec(\mathsf{sk}_{[0.\mathsf{vk}_\mathsf{s}]}, C_0) &= \mathcal{D}ec(\mathsf{sk}_{[0.\mathsf{vk}_\mathsf{s}]}, \mathcal{E}nc(\mathsf{pk}, [0.\mathsf{vk}_\mathsf{s}], M)) \\ &\to M' = M\end{aligned}$$

and

$$\begin{aligned}\mathcal{D}ec(\mathsf{sk}_{[1.\mathsf{vk}_\mathsf{s}]}, C_1) &= \mathcal{D}ec(\mathsf{sk}_{[1.\mathsf{vk}_\mathsf{s}]}, \mathcal{E}nc(\mathsf{pk}, [1.\mathsf{vk}_\mathsf{s}], H(M))) \\ &\to h' = H(M)\end{aligned}$$

where $\mathsf{sk}_{[i.\mathsf{vk}_\mathsf{s}]}$ is an outcome of $\mathcal{K}ey\mathcal{E}xt(\mathsf{sk}, [i.\mathsf{vk}_\mathsf{s}])$ for $i = 0, 1$. So, it holds $h' = H(M')$. In addition, because Sig is correct,

$$\mathsf{V}(\mathsf{vk}_\mathsf{s}, [C_0 \| C_1], \sigma) = \mathsf{V}(\mathsf{vk}_\mathsf{s}, [C_0 \| C_1], \mathsf{S}(\mathsf{sk}_\mathsf{s}, [C_0 \| C_1])) \to 1.$$

Hence, $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ always outputs the correct message M.

Let $\mathsf{ct}_k = (\mathsf{vk}_{\mathsf{s},k}, C_{k,0}, C_{k,1}, \sigma_k)$ be a valid ciphertext of user U_k, obtained by performing $\mathsf{Enc}(\mathsf{pk}_k, M_k)$ where pk_k is a user U_k's public key generated by running $\mathsf{KeyGen}(\mathsf{params})$ and params is an outcome of the Setup algorithm for $k = i, j$. Then, because $\mathcal{HIBE}$ is correct,

$$\mathcal{D}ec(\mathsf{sk}_{k,[1.\mathsf{vk}_{\mathsf{s},k}]}, C_{k,1}) \to h'_k = H(M_k)$$

where $\mathsf{sk}_{k,[1.\mathsf{vk}_{\mathsf{s},k}]}$ is an outcome of $\mathcal{K}ey\mathcal{E}xt(\mathsf{sk}_k, [1.\mathsf{vk}_{\mathsf{s},k}])$ for $k = i, j$. Hence, if $M_i = M_j$, then $h'_i = H(M_i) = H(M_j) = h'_j$ and so the Test algorithm always outputs 1. On the other hand, if $\mathsf{Dec}(\mathsf{sk}_i, \mathsf{ct}_i) \neq \mathsf{Dec}(\mathsf{sk}_j, \mathsf{ct}_j)$, then $M_i \neq M_j$ and so $h'_i = H(M_i) \neq H(M_j) = h'_j$ with overwhelming probability because H is collision resistant. Hence, the Test algorithm outputs 0 with overwhelming probability.

From the above, it is proved that our proposed construction is correct. □

5 Security Analysis

In this section, we look into the security of our PKEET construction against Type-I and Type-II adversaries. We first show that our scheme is IND-CCA2 secure against Type-II adversaries who do not have a trapdoor for equality tests on all target user's ciphertexts.

Theorem 2 (IND-CCA2). *If $\mathcal{HIBE}$ is an IND-sID-CPA secure 2-level HIBE scheme and Sig is a strongly unforgeable one-time signature scheme, then the proposed PKEET scheme exploiting $\mathcal{HIBE}$ and Sig is IND-CCA2 secure against Type-II adversaries in the standard model.*

More precisely, if there is no PPT adversary that breaks the strong unforgeability of one-time signature scheme Sig with at least $\varepsilon_{\mathsf{Sig}}$ success probability and there is no PPT adversary that breaks the IND-sID-CPA security of $\mathcal{HIBE}$ with at least $\varepsilon_{\mathcal{HIBE}}$ advantage, then for any PPT adversary that breaks the IND-CCA2 security of the proposed PKEET construction, its advantage is bounded above by $2\varepsilon_{\mathcal{HIBE}} + \dfrac{3\varepsilon_{\mathsf{Sig}}}{2}$.

Proof. We prove by using the standard hybrid argument. Let N be the number of users in the system and t be the index of the target user. Let $\mathsf{ct}_t^* = (\mathsf{vk_s}{}_{,t}^*, C_{t,0}^*, C_{t,1}^*, \sigma_t^*)$ be the challenge ciphertext for the user U_t.

We begin with defining the following three games and denote by $\mathcal{G}_i$ the event that the adversary $\mathcal{A}$ wins in $\mathbf{Game}_i$.

$\mathbf{Game}_0$: This game is the same as the original IND-CCA2 security game in Definition 9.

$\mathbf{Game}_1$: This game is almost the same as $\mathbf{Game}_0$, except that if $\mathcal{A}$ queries the oracle $\mathcal{O}^{\mathsf{Dec}}(t,\cdot)$ on $\mathsf{ct}_t = (\mathsf{vk}_{\mathsf{s},t}, C_{t,0}, C_{t,1}, \sigma_t)$ such that $\mathsf{vk}_{\mathsf{s},t} = \mathsf{vk_s}{}_{,t}^*$, $\mathsf{ct}_t \neq \mathsf{ct}_t^*$, and $\mathsf{V}(\mathsf{vk}_{\mathsf{s},t}, [C_{t,0} \| C_{t,1}], \sigma_t) \to 1$, then the challenger $\mathcal{C}$ stops an interaction and sets $\mathcal{A}$'s answer at random.

$\mathbf{Game}_2$: This game is almost the same as $\mathbf{Game}_1$, except for the challenger's response in the challenge phase. Recall that the original challenge ciphertext ct_t^* (in $\mathbf{Game}_0$ and $\mathbf{Game}_1$) has the form $(\mathsf{vk_s}{}_{,t}^*, C_{t,0}^*, C_{t,1}^*, \sigma_t^*)$, where $\mathsf{G}(\lambda) \to (\mathsf{vk_s}{}_{,t}^*, \mathsf{sk_s}{}_{,t}^*)$, $\mathcal{E}nc(\mathsf{pk}_t, [0.\mathsf{vk_s}{}_{,t}^*], M_b) \to C_{t,0}^*$, $\mathcal{E}nc(\mathsf{pk}_t, [1.\mathsf{vk_s}{}_{,t}^*], H(M_b)) \to C_{t,1}^*$, $\mathsf{S}(\mathsf{sk_s}{}_{,t}^*, [C_{t,0}^* \| C_{t,1}^*]) \to \sigma_t^*$, and b is a random bit chosen by $\mathcal{C}$. At the beginning of the challenge phase, $\mathcal{A}$ issues two messages M_0 and M_1. Then, $\mathcal{C}$ tosses two unbiased coins a and b. If $a = 0$, $\mathcal{C}$ computes the challenge ciphertext by using M_b and $H(M_{1-b})$, instead of M_b and $H(M_b)$. Otherwise ($a = 1$), $\mathcal{C}$ uses M_{1-b} and $H(M_b)$ for generating C_0 and C_1, respectively.

Let the advantage of $\mathcal{A}$ in $\mathbf{Game}_i$ be ε_i for $i = 0, 1, 2$. That is, $\varepsilon_i = \left| \Pr[\mathcal{G}_i] - \dfrac{1}{2} \right|$ for $i = 0, 1, 2$. Then, we prove the theorem by using a sequence of lemmas.

Lemma 1. $\varepsilon_0 - \varepsilon_1 < \frac{3\varepsilon_{\mathsf{Sig}}}{2}$.

Proof. We begin with defining an event E_1. In $\mathbf{Game}_1$, $\mathcal{C}$ should stop and output a random guess for $\mathcal{A}$'s output if $\mathcal{A}$ issues a decryption query on the target user U_t's ciphertext $\mathsf{ct}_t = (\mathsf{vk}_{\mathsf{s},t}, C_{t,0}, C_{t,1}, \sigma_t)$ such that $\mathsf{vk}_{\mathsf{s},t} = \mathsf{vk}_{\mathsf{s},t}^*$, $\mathsf{ct}_t \neq \mathsf{ct}_t^*$, and $\mathsf{V}(\mathsf{vk}_{\mathsf{s},t}, [C_{t,0} \| C_{t,1}], \sigma_t) \to 1$. Denote such the situation by E_1.

From the definitions of E_1, ε_0, and ε_1, we find a relation among $\Pr[E_1]$, ε_0, and ε_1 as follows.

$$\begin{aligned}
\varepsilon_1 &= \left| \Pr[\mathcal{G}_1] - \frac{1}{2} \right| \\
&= \left| \Pr[\mathcal{G}_1 | E_1] \Pr[E_1] + \Pr[\mathcal{G}_1 | \neg E_1] \Pr[\neg E_1] - \frac{1}{2} \right| \\
&= \left| \frac{1}{2} \Pr[E_1] + \Pr[\mathcal{G}_0 \wedge \neg E_1] - \frac{1}{2} \right| \\
&= \left| \frac{1}{2} \Pr[E_1] + \Pr[\mathcal{G}_0] - \Pr[\mathcal{G}_0 \wedge E_1] - \frac{1}{2} \right| \\
&\geq \left| \Pr[\mathcal{G}_0] - \frac{1}{2} \right| - \left| \frac{1}{2} \Pr[E_1] - \Pr[\mathcal{G}_0 \wedge E_1] \right| \\
&\geq \left| \Pr[\mathcal{G}_0] - \frac{1}{2} \right| - \frac{3}{2} \Pr[E_1] \\
&= \varepsilon_0 - \frac{3}{2} \Pr[E_1].
\end{aligned}$$

The third equality holds since $\Pr[\mathcal{G}_1 | E_1] = \dfrac{1}{2}$ and $\Pr[\mathcal{G}_0 \wedge \neg E_1] = \Pr[\mathcal{G}_1 \wedge \neg E_1]$. Hence, we have

$$\Pr[E_1] \geq \frac{2}{3}(\varepsilon_0 - \varepsilon_1). \tag{1}$$

Now, we are going to show that $\Pr[E_1] < \varepsilon_{\mathsf{Sig}}$, which leads an inequality among ε_0, ε_1, and $\varepsilon_{\mathsf{Sig}}$. In $\mathbf{Game}_1$, what $\mathcal{A}$ causes the event E_1 is that $\mathcal{A}$ succeeds in forging a signature on some message with respect to the verification key $\mathsf{vk}_{\mathsf{s},t}^*$. Here, $\mathsf{vk}_{\mathsf{s},t}^*$ appears only in the challenge ciphertext. So, one can easily embed an instance of the strong unforgeability game for the underlying signature scheme Sig into $\mathbf{Game}_1$; that is, one can construct another simulator $\mathcal{S}_{\mathsf{Sig}}$ that breaks the strong unforgeability of Sig by using $\mathcal{A}$ in $\mathbf{Game}_1$. One lets $\mathcal{S}_{\mathsf{Sig}}$ behave normally as the challenger of $\mathbf{Game}_1$ for all situations, except for the challenge ciphertext generation. Once $\mathcal{S}_{\mathsf{Sig}}$ generates the challenge ciphertext, it embeds the verification key of the strong unforgeability game for Sig as $\mathsf{vk}_{\mathsf{s},t}^*$ in the challenge ciphertext. It normally generates ciphertexts $C_{t,0}^*$ and $C_{t,1}^*$ of a message M_b and its hash value $H(M_b)$, respectively, by selecting $b \in \{0,1\}$ at random. It then requests a signature of the message $[C_{t,0}^* \| C_{t,1}^*]$ to the signing oracle that is given to $\mathcal{S}_{\mathsf{Sig}}$ in the strong unforgeability game. If the event E_1 occurs, what $\mathcal{A}$ returns is exactly a forged signature on some message, which is different from $[C_{t,0}^* \| C_{t,1}^*]$, with respect to the target verification key $\mathsf{vk}_{\mathsf{s},t}^*$. Hence, $\Pr[E_1]$ should be less than $\varepsilon_{\mathsf{Sig}}$ by the assumption that there is no PPT adversary that breaks the strong unforgeability of Sig with at least $\varepsilon_{\mathsf{Sig}}$ success probability. Finally, putting this together with Equation (1), we have the following inequality stated in the lemma,

$$\varepsilon_0 - \varepsilon_1 < \frac{3}{2}\varepsilon_{\mathsf{Sig}}. \tag{2}$$

□

Lemma 2. $\varepsilon_1 - \varepsilon_2 < 2\varepsilon_{\mathcal{HIBE}}$.

Proof. We construct a simulator $\mathcal{S}_{\mathcal{HIBE}}$ that breaks the IND-sID-CPA security of $\mathcal{HIBE}$ by using $\mathcal{A}$. Denote the challenger of the IND-sID-CPA security game for $\mathcal{HIBE}$ by $\mathcal{C}_{\mathcal{HIBE}}$. First, we describe $\mathcal{S}_{\mathcal{HIBE}}$ and then analyze its advantage later. $\mathcal{S}_{\mathcal{HIBE}}$ generates a pair of a signing key and a verification key $(\mathsf{sk_s}{}^*_{,t}, \mathsf{vk_s}{}^*_{,t})$ by running the key generation algorithm G of the signature scheme Sig. $\mathcal{S}_{\mathcal{HIBE}}$ tosses a coin $\alpha \in \{0,1\}$ and passes $[\alpha.\mathsf{vk_s}{}^*_{,t}]$ as the target (2-level) identity to $\mathcal{C}_{\mathcal{HIBE}}$. Once receiving the master public key mpk of $\mathcal{HIBE}$ from $\mathcal{C}_{\mathcal{HIBE}}$, $\mathcal{S}_{\mathcal{HIBE}}$ sets a user U_t's public key pk_t to mpk, runs the key extraction algorithm $\mathcal{S}etup(\lambda) \rightarrow (\mathsf{pk}_i, \mathsf{sk}_i)$ for all $1 \leq i \neq t \leq N$, and passes the system parameter $\mathsf{params} = \{H, \mathcal{HIBE}, \mathsf{Sig}\}$ and all pk_i's for $1 \leq i \leq N$ to $\mathcal{A}$.

As for the query responses, $\mathcal{S}_{\mathcal{HIBE}}$ utilizes her own oracles provided by $\mathcal{C}_{\mathcal{HIBE}}$. If $\mathcal{A}$ issues a decryption query on a valid ciphertext using $\mathsf{vk_s}{}^*_{,t}$, then $\mathcal{S}_{\mathcal{HIBE}}$ stops the interaction with $\mathcal{A}$ and sets $\mathcal{A}$'s output to be a random coin. For all the other decryption queries to $\mathcal{O}^{\mathsf{Dec}}$, $\mathcal{S}_{\mathcal{HIBE}}$ responds by accessing the key extraction oracle served by $\mathcal{C}_{\mathcal{HIBE}}$. For all other queries $\mathcal{O}^{sk}(i)$ and $\mathcal{O}^{\mathsf{Td}}(i)$ with $i \neq t$, $\mathcal{S}_{\mathcal{HIBE}}$ responds by using sk_i's. We note that $\mathcal{A}$ will not issue a secret key extraction query $\mathcal{O}^{sk}(t)$ and a trapdoor query $\mathcal{O}^{\mathsf{Td}}(t)$ for the target user U_t in the IND-CCA2 security game.

In the challenge phase, $\mathcal{A}$ chooses and sends M_0 and M_1 to $\mathcal{S}_{\mathcal{HIBE}}$. According to α, which is chosen by $\mathcal{S}_{\mathcal{HIBE}}$ at the beginning of the simulation, the challenge ciphertext for $\mathcal{A}$ is differently computed: If $\alpha = 0$, then $\mathcal{S}_{\mathcal{HIBE}}$ sends M_0 and M_1 to $\mathcal{C}_{\mathcal{HIBE}}$, receives $C^*_{t,0,b}$ that is a ciphertext of the message M_b with respect to the target identity $[0.\mathsf{vk_s}{}^*]$, where b is a random bit chosen by $\mathcal{C}_{\mathcal{HIBE}}$. Then, $\mathcal{S}_{\mathcal{HIBE}}$ selects a random bit $\beta \in \{0,1\}$ and sends $\mathcal{A}$

$$\mathsf{ct}^*_t = (\mathsf{vk_s}{}^*_{,t}, C^*_{t,0,b}, C^*_{t,1,\beta} = \mathcal{E}nc(\mathsf{pk}_t, [1.\mathsf{vk_s}{}^*_{,t}], H(M_\beta)), \sigma^*_t),$$

where $\mathsf{S}(\mathsf{sk_s}{}^*_{,t}, [C^*_{t,0,b} \| C^*_{t,1,\beta}]) \rightarrow \sigma^*_t$. Similarly, if $\alpha = 1$, then $\mathcal{S}_{\mathcal{HIBE}}$ sends $H(M_0)$ and $H(M_1)$ to $\mathcal{C}_{\mathcal{HIBE}}$, and receives $C^*_{t,1,b}$ that is a ciphertext of the message $H(M_b)$ with respect to the target identity $[1.\mathsf{vk_s}{}^*_{,t}]$, where b is a random bit chosen by $\mathcal{C}_{\mathcal{HIBE}}$. Then, $\mathcal{S}_{\mathcal{HIBE}}$ selects a random bit $\beta \in \{0,1\}$ and sends $\mathcal{A}$

$$\mathsf{ct}^*_t = (\mathsf{vk_s}{}^*_{,t}, C^*_{t,0,\beta} = \mathcal{E}nc(\mathsf{pk}_t, [0.\mathsf{vk_s}{}^*_{,t}], M_\beta), C^*_{t,1,b}, \sigma^*_t),$$

where $\mathsf{S}(\mathsf{sk_s}{}^*_{,t}, [C^*_{t,0,\beta} \| C^*_{t,1,b}]) \rightarrow \sigma^*_t$. After the challenge phase, $\mathcal{S}_{\mathcal{HIBE}}$ can respond to all decryption queries correctly as before the challenge phase. Finally, $\mathcal{S}_{\mathcal{HIBE}}$ forwards $\mathcal{A}$'s output b' to $\mathcal{C}_{\mathcal{HIBE}}$.

From now on, let us analyze our simulation $\mathcal{S}_{\mathcal{HIBE}}$. It is trivial to show that all the simulated transcripts are identical to the viewpoint of $\mathcal{A}$ in **Game**$_1$, except the challenge ciphertext. As for the challenge ciphertext, if $b = \beta$, the challenge ciphertext has the same form as that of $\mathbf{Game}_1$. Otherwise (that is, $b \neq \beta$), it is the challenge ciphertext of $\mathbf{Game}_2$. Hence, the advantage of $\mathcal{S}_{\mathcal{HIBE}}$ can be computed as follows.

$$\begin{aligned}
\varepsilon_{\mathcal{HIBE}} > {} & \mathbf{Adv}^{\text{IND-sID-CPA}}_{\mathcal{S}_{\mathcal{HIBE}},\mathcal{HIBE}}(\lambda) \\
= {} & \Big| \Pr[b' = b] - \frac{1}{2} \Big| \\
= {} & \Big| \frac{1}{2}\big(\Pr[b' = b | b = \beta] + \Pr[b' = b | b \neq \beta] \big) - \frac{1}{2} \Big| \\
= {} & \frac{1}{2} \Big| \Pr[\mathcal{G}_1] + \Pr[\mathcal{G}_2] - 1 \Big| \\
= {} & \frac{1}{2} \Big| \big(\Pr[\mathcal{G}_1] - \frac{1}{2} \big) + \big(\Pr[\mathcal{G}_2] - \frac{1}{2} \big) \Big| \\
\geq {} & \frac{1}{2} \Big(\Big| \Pr[\mathcal{G}_1] - \frac{1}{2} \Big| - \Big| \Pr[\mathcal{G}_2] - \frac{1}{2} \Big| \Big) \\
= {} & \frac{1}{2}(\varepsilon_1 - \varepsilon_2).
\end{aligned}$$

□

Lemma 3. $\varepsilon_2 = 0$.

Proof. In the challenge phase, $\mathcal{C}$ computes the challenge ciphertext differently according to the choice of a. If $a = 0$, $\mathcal{C}$ uses M_b and $H(M_{1-b})$ for generating C_0 and C_1, respectively. Otherwise ($a = 1$), $\mathcal{C}$ uses M_{1-b} and $H(M_b)$. Even when messages are known to $\mathcal{A}$, $\mathcal{A}$ cannot find the value b correctly since a is completely hidden from the viewpoint of $\mathcal{A}$. Hence, $\varepsilon_2 = 0$. □

Overall, putting all the results of lemmas, we have

$$\varepsilon_0 \leq 2\varepsilon_{\mathcal{HIBE}} + \frac{3\varepsilon_{\mathsf{Sig}}}{2}.$$

□

The following theorem shows that our PKEET construction is OW-CCA2 secure against Type-I adversaries who have a trapdoor for equality tests on all target user's ciphertexts.

Theorem 3 (OW-CCA2). *If $\mathcal{HIBE}$ is an IND-sID-CPA secure 2-level HIBE scheme, H is a one-way hash function, and Sig is a strongly unforgeable one-time signature scheme, then the proposed PKEET scheme exploiting H, $\mathcal{HIBE}$, and Sig, is OW-CCA2 secure against Type-I adversaries in the standard model.*

More precisely, if there is no PPT adversary that breaks one of the strong unforgeability of Sig, the one-wayness of H, and the IND-sID-CPA security of $\mathcal{HIBE}$ with at least $\varepsilon_{\mathsf{Sig}}$ success probability, ε_H success probability, and $\varepsilon_{\mathcal{HIBE}}$ advantage, respectively, then for any PPT adversary that breaks the OW-CCA2 security of the proposed PKEET, its success probability is bounded above by $4\varepsilon_{\mathcal{HIBE}} + \varepsilon_H + \varepsilon_{\mathsf{Sig}}$.

Proof. Similarly to the IND-CCA2 security proof of Theorem 2, we use the standard hybrid argument and begin with defining two games. Let N be the number of users in the system and t be the index of the target user. Let $\mathsf{ct}_t^* = (\mathsf{vk_s}_{,t}^*, C_{t,0}^*, C_{t,1}^*, \sigma_t^*)$ be the challenge ciphertext for the target user U_t. Denote by $\mathcal{G}_i$ the event that the adversary $\mathcal{A}$ wins in $\mathbf{Game}_i$.

$\mathbf{Game}_0$: This game is the same as the original OW-CCA2 security game in Definition 8.

$\mathbf{Game}_1$: This game is almost the same as $\mathbf{Game}_0$, except that if $\mathcal{A}$ queries $\mathcal{O}^{\mathsf{Dec}}(t, \cdot)$ on $\mathsf{ct}_t = (\mathsf{vk}_{\mathsf{s},t}, C_{t,0}, C_{t,1}, \sigma_t)$ such that $\mathsf{vk}_{\mathsf{s},t} = \mathsf{vk_s}_{,t}^*$, $\mathsf{ct}_t \neq \mathsf{ct}_t^*$, and $\mathsf{V}(\mathsf{vk}_{\mathsf{s},t}, [C_{t,0} \| C_{t,1}], \sigma_t) \to 1$, then the challenger $\mathcal{C}$ stops an interaction and sets $\mathcal{A}$'s answer at random.

We show that the adversarial success probability gap between the above two games is less than $\varepsilon_{\mathsf{Sig}}$. In $\mathbf{Game}_1$, $\mathcal{C}$ should stop and output a random guess if $\mathcal{A}$ issues a decryption query on a ciphertext $\mathsf{ct}_t = (\mathsf{vk}_{\mathsf{s},t}, C_{t,0}, C_{t,1}, \sigma_t)$ such that $\mathsf{vk}_{\mathsf{s},t} = \mathsf{vk_s}_{,t}^*$, $\mathsf{ct}_t \neq \mathsf{ct}_t^*$, and $\mathsf{V}(\mathsf{vk}_{\mathsf{s},t}, [C_{t,0} \| C_{t,1}], \sigma_t) \to 1$. Denote such the situation by E_1. We assume that there exists an adversary $\mathcal{A}$ such that its success probability in $\mathbf{Game}_i$ is ε_i for $i = 0, 1$. That is, $\varepsilon_i = \Pr[\mathcal{G}_i]$ for $i = 0, 1$. Then, we find a relation among $\Pr[E_1]$, ε_0, and ε_1 as follows:

$$\begin{aligned}
\varepsilon_1 &= \Pr[\mathcal{G}_1] \\
&= \Pr[\mathcal{G}_1 | E_1] \Pr[E_1] + \Pr[\mathcal{G}_1 | \neg E_1] \Pr[\neg E_1] \\
&= \frac{1}{|\mathcal{M}|} \Pr[E_1] + \Pr[\mathcal{G}_0 \wedge \neg E_1] \\
&\geq \frac{1}{|\mathcal{M}|} \Pr[E_1] + \Pr[\mathcal{G}_0] - \Pr[E_1] \\
&\geq \varepsilon_0 - \Pr[E_1].
\end{aligned}$$

Hence, we have

$$\Pr[E_1] \geq \varepsilon_0 - \varepsilon_1. \tag{3}$$

We can show that $\Pr[E_1] < \varepsilon_{\mathsf{Sig}}$ by using the same argument given in the proof of Theorem 2 and so we omit the details. Therefore, we finally have an inequality

$$\varepsilon_0 - \varepsilon_1 < \varepsilon_{\mathsf{Sig}}. \tag{4}$$

Next, we compute the success probability of $\mathcal{A}$ in $\mathbf{Game}_1$. To this end, we first analyze $\mathcal{A}$'s behaviour on anomalous inputs, which will be used for analysis of our main simulation. We call this *pre*-simulation $\mathcal{PS}$. In $\mathcal{PS}$, the overall role of the simulator is exactly the same as a normal challenger in $\mathbf{Game}_1$, except the challenge ciphertext. For the challenge ciphertext, the simulator chooses two different messages M_0' and M_1' at random, generates ciphertexts $C_{t,0}'^*$ and $C_{t,1}'^*$ of messages M_0' and $H(M_1')$, respectively. All other steps for generating the challenge ciphertext are the same as those of the normal challenger. Here, we cannot expect how $\mathcal{A}$ behaves on this anomalous transcript, but we can find an upper bound of the probability that $\mathcal{A}$ outputs M'_1 by considering another game to break the one-wayness of H using $\mathcal{A}$. The reduction is quite straightforward. One can just embed the instance of the one-wayness game into a message of $C'^*_{t,1}$, instead of $H(M'_1)$. Since M'_1 is used only for $C'^*_{t,1}$ in the challenge ciphertext, the simulation for the other parts is straightforward. From this simulation, we have

$$\Pr[\mathcal{A} \to M'_1 \text{ in } \mathcal{PS}] < \varepsilon_H. \tag{5}$$

Now, we are ready to construct a main simulator $\mathcal{S}_{\mathcal{HIBE}}$ that breaks the IND-sID-CPA security of $\mathcal{HIBE}$ by using $\mathcal{A}$. Denote the challenger of the IND-sID-CPA security game for $\mathcal{HIBE}$ by $\mathcal{C}_{\mathcal{HIBE}}$. First, $\mathcal{S}_{\mathcal{HIBE}}$ chooses a target verification key $\mathsf{vk_s}^*_{,t}$ by running $\mathsf{G}(\lambda) \to (\mathsf{vk_s}^*_{,t}, \mathsf{sk_s}^*_{,t})$. Next, $\mathcal{S}_{\mathcal{HIBE}}$ sends $[0.\mathsf{vk_s}^*]$ as the target (2-level) identity to $\mathcal{C}_{\mathcal{HIBE}}$. Once $\mathcal{S}_{\mathcal{HIBE}}$ receives the system parameter mpk of $\mathcal{HIBE}$, it sets pk_t to mpk, runs the key extraction algorithm $\mathcal{S}etup(\lambda) \to (\mathsf{pk}_i, \mathsf{sk}_i)$ for all $1 \le i \ne t \le N$, and passes the system parameter $\mathsf{params} = \{H, \mathcal{HIBE}, \mathsf{Sig}\}$ and all pk_i's for $1 \le i \le N$ to $\mathcal{A}$. $\mathcal{S}_{\mathcal{HIBE}}$ can respond to all decryption oracle queries, secret key extraction queries, and trapdoor queries correctly as in the proof of Theorem 2. In particular, differently from the adversary in the proof of Theorem 2, it is allowed that $\mathcal{A}$ issues a trapdoor query on the index of the target user U_t and $\mathcal{S}_{\mathcal{HIBE}}$ can answer by querying on the secret key for the identity 1 to the key extraction oracle of $\mathcal{HIBE}$.

In the challenge phase, $\mathcal{S}_{\mathcal{HIBE}}$ chooses M_0 and M_1 at random, sends them to $\mathcal{C}_{\mathcal{HIBE}}$, and then receives $C^*_{t,0,b}$ that is a ciphertext of the message M_b with respect to the target identity $[0.\mathsf{vk_s}^*]$, where b is a random bit chosen by $\mathcal{C}_{\mathcal{HIBE}}$. $\mathcal{S}_{\mathcal{HIBE}}$ selects a random bit $\beta \in \{0,1\}$ and sends $\mathcal{A}$

$$\mathsf{ct}^*_t = (\mathsf{vk_s}^*_{,t}, C^*_{t,0,b}, C^*_{t,1,\beta} = \mathcal{E}nc(\mathsf{pk}_t, [1.\mathsf{vk_s}^*_{,t}], H(M_\beta)), \sigma^*_t),$$

where $\mathsf{S}(\mathsf{sk_s}^*_{,t}, [C^*_{t,0,b} \| C^*_{t,1,\beta}]) \to \sigma^*_t$. After the challenge phase, $\mathcal{S}_{\mathcal{HIBE}}$ can respond to all queries correctly as before the challenge phase. Finally, $\mathcal{A}$ may output either a message M' or $\perp$ to $\mathcal{S}_{\mathcal{HIBE}}$. If $M' = M_\beta$, then $\mathcal{S}_{\mathcal{HIBE}}$ returns $b' = \beta$ to $\mathcal{C}_{\mathcal{HIBE}}$. Otherwise, $\mathcal{S}_{\mathcal{HIBE}}$ sets a bit b' at random and returns it to $\mathcal{C}_{\mathcal{HIBE}}$.

It is trivial to show that all the simulated transcripts are identical to the viewpoint of $\mathcal{A}$ in $\mathbf{Game}_1$, except the challenge ciphertext. If $b = \beta$, the simulated transcripts are identical to the real transcripts including the challenge ciphertext. Otherwise, both are not identical and hence we cannot expect $\mathcal{A}$'s behaviour exactly. However, we can obtain some probability about $\mathcal{A}$'s output in this case from the result of the simulation $\mathcal{PS}$. By using this, we compute the advantage of $\mathcal{S}_{\mathcal{HIBE}}$ as follows.

$$\begin{aligned}
&\Big|\Pr[b' = b] - \frac{1}{2}\Big| \\
= &\Big|\frac{1}{2}\big(\Pr[b' = b | b = \beta] + \Pr[b' = b | b \ne \beta]\big) - \frac{1}{2}\Big| \\
= &\Big|\frac{1}{2}\big(\Pr[\mathcal{A} \to M_b \vee (\mathcal{A} \not\to M_b \wedge b' = b) | b = \beta] + \Pr[b' = b | b \ne \beta]\big) - \frac{1}{2}\Big|
\end{aligned}$$

$$
\begin{aligned}
&= \Big|\frac{1}{2}\big(\Pr[\mathcal{A} \to M_b | b = \beta] + \Pr[(\mathcal{A} \not\to M_b \wedge b' = b) | b = \beta] + \Pr[b' = b | b \neq \beta]\big) - \frac{1}{2}\Big| \\
&= \Big|\frac{1}{2}\big(\Pr[\mathcal{G}_1] + \frac{1}{2}\Pr[\mathcal{A} \not\to M_b | b = \beta] + \Pr[b' = b | b \neq \beta]\big) - \frac{1}{2}\Big| \\
&= \Big|\frac{1}{2}\big(\Pr[\mathcal{G}_1] + \frac{1}{2}(1 - \Pr[\mathcal{G}_1]) + \Pr[b' = b | b \neq \beta]\big) - \frac{1}{2}\Big| \\
&= \frac{1}{2}\Big|\frac{1}{2}\Pr[\mathcal{G}_1] + \Pr[b' = b | b \neq \beta] - \frac{1}{2}\Big| \\
&= \frac{1}{2}\Big|\frac{1}{2}\Pr[\mathcal{G}_1] + \Pr[\mathcal{A} \not\to M_\beta \wedge b' = b | b \neq \beta] - \frac{1}{2}\Big| \\
&= \frac{1}{2}\Big|\frac{1}{2}\Pr[\mathcal{G}_1] + \frac{1}{2}\Pr[\mathcal{A} \not\to M_\beta | b \neq \beta] - \frac{1}{2}\Big| \\
&= \frac{1}{2}\Big|\frac{1}{2}\Pr[\mathcal{G}_1] + \frac{1}{2}(1 - \Pr[\mathcal{A} \to M_\beta | b \neq \beta]) - \frac{1}{2}\Big| \\
&= \frac{1}{4}\Big|\Pr[\mathcal{G}_1] - \Pr[\mathcal{A} \to M_\beta | b \neq \beta]\Big| \\
&> \frac{1}{4}(\Pr[\mathcal{G}_1] - \varepsilon_H)
\end{aligned}
$$

where $\mathcal{G}_1$ denotes the event that $\mathcal{A}$ wins in $\mathbf{Game}_1$. The fourth and eighth equalities hold because b' is independently and randomly chosen from the set $\{0, 1\}$ if $M' \neq M_\beta$ for $\mathcal{A}$'s output M' at the last step and the last inequality holds because of Equation (5).

Hence, we have $\frac{1}{4}(\varepsilon_1 - \varepsilon_H) < \varepsilon_{\mathcal{HIBE}}$. Therefore, putting this with Equation (4), we obtain $\varepsilon_0 < 4\varepsilon_{\mathcal{HIBE}} + \varepsilon_H + \varepsilon_{\mathsf{Sig}}$. □

6 Discussion

In this section, we provide an extension of our PKEET construction to the identity-based setting. We also present a comparison of ours with the previous results.

6.1 Extension to IBE with Equality Test

Because our PKEET construction already exploits a 2-level HIBE scheme, one can easily extend it to the identity-based setting by just employing a 3-level HIBE scheme for the underlying HIBE scheme. In our transformation to the identity-based setting, the first level of the underlying HIBE scheme is utilized for a user's identity ID, and the second and third levels are reserved for the roles of the first and second levels in our PKEET construction, respectively. Security analysis for our IBEET construction is almost the same as that of our PKEET construction, provided in Section 5. The main difference is the security requirement of the underlying HIBE scheme: While the IND-sID-CPA security is sufficient for our PKEET construction, the IND-ID-CPA security is required for the identity-based setting. As a result, we obtain the first IBEET construction that achieves one-wayness under adaptive identity and adaptive chosen ciphertext attacks (OW-ID-CCA2) against Type-I adversaries and achieves the

Table 1. Comparison of Our PKEET with Existing Schemes

		[23]	[22]	[16]	Ours (with [5] +[7])
Comp of	Enc	$3\mathsf{Exp}$	$5\mathsf{Exp}$	$6\mathsf{Exp}$	$1\mathsf{Pairing} + 14\mathsf{Exp}$
	Dec	$3\mathsf{Exp}$	$2\mathsf{Exp}$	$5\mathsf{Exp}$	$9\mathsf{Pairing} + 11\mathsf{Exp}$
	Test	$2\mathsf{Pairing}$	$4\mathsf{Exp}$	$2\mathsf{Pairing} + 2\mathsf{Exp}$	$6\mathsf{Pairing} + 10\mathsf{Exp}$
Size of	PK	$\|\mathbb{G}\|$	$2\|\mathbb{G}\|$	$3\|\mathbb{G}\|$	$5\|\mathbb{G}\|$
	CT	$3\|\mathbb{G}\| + \|\mathbb{Z}_p\|$	$4\|\mathbb{G}\| + \|\mathbb{Z}_p\| + 2\lambda$	$5\|\mathbb{G}\| + \|\mathbb{Z}_p\|$	$(2\lambda + 15)\|\mathbb{G}\| + \|\mathbb{Z}_p\|$
	TD	–	$\|\mathbb{Z}_p\|$	$\|\mathbb{Z}_p\|$	$3\|\mathbb{G}\|$
Security	Type-I	OW-CCA2	OW-CCA2	OW-CCA2	OW-CCA2
	Type-II	–	IND-CCA2	IND-CCA2	IND-CCA2
Standard Model		No	No	No	Yes
Assumptions		CDH	CDH	CDH	CDH+DBDH

Legends: Comp: computational complexity, Enc: encryption algorithm, Dec: decryption algorithm, Test: test algorithm, PK: public key, CT: ciphertext, TD: trapdoor, Exp: cost for an exponentiation, $\mathsf{Pairing}$: cost for a pairing computation, λ: security parameter, CDH: computational Diffie-Hellman assumption, DBDH: decisional Diffie-Hellman assumption

indistinguishability under adaptive identity and adaptive chosen ciphertext attacks (IND-ID-CCA2) against Type-II adversaries in the standard model. We provide the details of the description of our IBEET construction and its security proofs in Appendix B.

6.2 Comparison of Our PKEET Construction with Previous Works

We provide a comparison of our PKEET construction with previous schemes. For our scheme, we exploit Boneh and Boyen's IND-sID-CPA HIBE (BB-HIBE) [5] and Boneh, Shen, and Waters' strongly unforgeable signature (BSW-Sig) [7] as underlying HIBE and signature schemes, respectively. We note that for a level-ℓ user, the BB-HIBE scheme requires $(2\ell + 1)\mathsf{Exp}$, $1\mathsf{Pairing} + (\ell + 2)\mathsf{Exp}$, and $(\ell + 1)\mathsf{Pairing}$ for key generation, encryption, and decryption, respectively, where $\mathsf{Pairing}$ and Exp denote costs for computing a pairing and an exponentiation, respectively. The sizes of a ciphertext and a private key for a level-ℓ user are $(2 + \ell)|\mathbb{G}|$ and $\ell|\mathbb{G}|$, respectively, where $|\mathbb{G}|$ denotes a bit size required for representing an element in the underlying group $\mathbb{G}$. The security of their scheme relies on decisional bilinear Diffie-Hellman (DBDH) assumptions. On the other hand, the BSW-Sig scheme requires $1\mathsf{Exp}$, $6\mathsf{Exp}$, and $3\mathsf{Pairing} + 1\mathsf{Exp}$ for key generation, signing, and verification, respectively. The bit sizes of a verification key and a signature are $(n + 5)|\mathbb{G}|$ and $2|\mathbb{G}| + |\mathbb{Z}_p|$, respectively, where n is the output size of the utilized hash function and p is the order of the underlying group $\mathbb{G}$. Its security relies on computational Diffie-Hellman (CDH) assumptions.

Table 1 presents a comparison of ours with previous works. The second, third, fourth, and last columns describe the features of Yang et al.'s original PKEET [23], Tang's all-or-nothing PKEET [22], Ma et al.'s PKEET [16] by considering a Type-1 authorization only in their paper, and ours, respectively. We set the output size of the utilized hash functions to 2λ for the security parameter λ. The table shows that ours has the worst performance among them in terms of both computational complexity and parameter size. The performance of ours in the table heavily relies on that of the underlying signature and HIBE schemes. Thus, we believe that more efficient underlying schemes improve the performance of ours further. On the other hand, we remark that our PKEET construction is the first one in the standard model.

7 Conclusion

In this paper, we provided a generic construction of PKEET by exploiting a 2-level HIBE scheme, a traditional signature scheme, and a cryptographic hash function. Our proposed scheme is OW-CCA2 secure against Type-I adversaries who have a trapdoor for equality tests and is IND-CCA2 secure against Type-II adversaries who do not have if the exploited HIBE scheme is IND-sID-CPA secure, the exploited one-time signature scheme is strongly unforgeable, and the exploited hash function is one-way in the standard model. As a result, we obtain the first PKEET construction that is secure in the standard model. Finally, we discussed an extension of our PKEET construction into the identity-based setting.

While our construction has an advantage that is generic in the sense that it does not require any number-theoretic assumption explicitly, there may be room for improvement in terms of efficiency. Thus, it would be interesting to design efficient PKEET schemes in the standard model.

Acknowledgements

Hyung Tae Lee would like to thank Khoa Nguyen for his vaulable suggstions to improve the previous version of this paper. Hyung Tae Lee, San Ling, and Huaxiong Wang were supported by Research Grant TL-9014101684-01 and the Singapore Ministry of Education under Research Grant MOE2013-T2-1-041. Huaxiong Wang was also supported by NTU under Tier 1 grant RG143/14. Jae Hong Seo was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [15ZS1500, Develpoment of storage and search technologies over encrypted database].

References

1. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency

properties, relation to anonymous IBE, and extensions. *J. Cryptology*, 21(3):350–391, 2008.

2. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In H. Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, 2010.
3. S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In T. Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*, pages 98–115. Springer, 2010.
4. M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, 2007.
5. D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.
6. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 506–522. Springer, 2004.
7. D. Boneh, E. Shen, and B. Waters. Strongly unforgeable signatures based on computational Diffie-Hellman. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, *Public Key Cryptography (PKC) 2006*, volume 3958 of *LNCS*, pages 229–240. Springer, 2006.
8. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). In J. S. Vitter, editor, *ACM STOC 1998*, pages 209–218. ACM, 1998.
9. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.
10. Y. Dodis, R. Oliveira, and K. Pietrzak. On the generic insecurity of the full domain hash. In V. Shoup, editor, *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *LNCS*, pages 449–466. Springer, 2005.
11. K. Huang, R. Tso, Y. Chen, S. M. M. Rahman, A. Almogren, and A. Alamri. PKE-AET: Public key encryption with authorized equality test. *Comput. J.*, 58(10):2686–2697, 2015.
12. H. T. Lee, S. Ling, J. H. Seo, and H. Wang. Semi-generic construction of public key encryption and identity-based encryption with equality test. *Inf. Sci.*, 373:419–440, 2016.
13. G. Leurent and P. Q. Nguyen. How risky is the random-oracle model? In S. Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 445–464. Springer, 2009.
14. X.-J. Lin, H. Qu, and X. Zhang. Public key encryption supporting equality test and flexible authorization without bilinear pairings. *IACR Cryptology ePrint Archive*, 2016:277, 2016.
15. S. Ma. Identity-based encryption with outsourced equality test in cloud computing. *Inf. Sci.*, 328:389–402, 2016.
16. S. Ma, Q. Huang, M. Zhang, and B. Yang. Efficient public key encryption with equality test supporting flexible authorization. *IEEE Transactions on Information Forensics and Security*, 10(3):458–470, 2015.
17. S. Ma, M. Zhang, Q. Huang, and B. Yang. Public key encryption with delegated equality test in a multi-user setting. *Comput. J.*, 58(4):986–1002, 2015.

18. M. Rückert. Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles. In N. Sendrier, editor, *Post-Quantum Cryptography (PQCrypt) 2010*, volume 6061 of *LNCS*, pages 182–200. Springer, 2010.
19. D. Slamanig, R. Spreitzer, and T. Unterluggauer. Adding controllable linkability to pairing-based group signatures for free. In S. S. M. Chow, J. Camenisch, L. C. K. Hui, and S. Yiu, editors, *Information Security Conference (ISC) 2014*, volume 8783 of *LNCS*, pages 388–400. Springer, 2014.
20. Q. Tang. Towards public key encryption scheme supporting equality test with fine-grained authorization. In U. Parampalli and P. Hawkes, editors, *ACISP 2011*, volume 6812 of *LNCS*, pages 389–406. Springer, 2011.
21. Q. Tang. Public key encryption schemes supporting equality test with authorisation of different granularity. *IJACT*, 2(4):304–321, 2012.
22. Q. Tang. Public key encryption supporting plaintext equality test and user-specified authorization. *Security and Communication Networks*, 5(12):1351–1362, 2012.
23. G. Yang, C. H. Tan, Q. Huang, and D. S. Wong. Probabilistic public key encryption with equality test. In J. Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010*, volume 5985 of *LNCS*, pages 119–131. Springer, 2010.

A Syntax and System Model for IBEET

In this section, we provide basic definitions of IBEET, including a system model for IBEET, a definition of an IBEET scheme, and its security definitions.

System Model for Our IBEET. Our IBEET system consists of multiple users (e.g., senders and receivers), the key generation center (KGC), and tester(s) (e.g., the server): As traditional IBE schemes, once a user requests his/her secret key to the KGC by sending his/her identity, the KGC issues a user's secret key according to the user's identity to the user. The rest is almost the same as that of our PKEET system model. A sender encrypts a data using a receiver's identity and sends a ciphertext to the receiver. The receiver may decrypt his/her ciphertexts using his/her secret key and/or store ciphertexts at the server. When the receiver wants to delegate the test capability for all of his/her ciphertexts, he/she issues a trapdoor for equality tests to a tester who can access the server that stores his/her ciphertexts. From that moment, the tester can perform equality tests on ciphertexts under the identity of the receiver who delegated the test authority to the tester.

Definitions of IBEET. Below, we present the formal definition of IBEET schemes under our system model.

Definition 10 (Identity-Based Encryption with Equality Test). *An identity-based encryption with equality test (IBEET) consists of the following six polynomial-time algorithms* $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Td}, \mathsf{Test})$*:*

– $\mathsf{Setup}(\lambda)$*: On input a security parameter* λ*, it outputs a public parameter* pp *and a master secret key* msk*. We note that* pp *includes the information of* *the message space $\mathcal{M}$ and it is assumed that all other algorithms take pp as an input implicitly, though it is not stated.*

- $\mathsf{Extract}(pp, msk, \mathrm{ID})$*: It takes pp, msk, and an identity* $\mathrm{ID} \in \{0,1\}^*$ *as inputs, and outputs a user* ID*'s secret key* d_{ID}.
- $\mathsf{Enc}(pp, \mathrm{ID}, M)$*: It takes pp, an identity* ID*, and a message* $M \in \mathcal{M}$ *as inputs and outputs a ciphertext* ct.
- $\mathsf{Dec}(pp, d_{\mathrm{ID}}, \mathsf{ct})$*: It takes pp, a user* ID*'s secret key* d_{ID} *and a ciphertext* ct *as inputs and outputs a message* M' *or* $\perp$.
- $\mathsf{Td}(d_{\mathrm{ID}})$*: On input a user* ID*'s secret key* d_{ID}*, it outputs a trapdoor* $\mathsf{td}_{\mathrm{ID}}$.
- $\mathsf{Test}(\mathsf{td}_{\mathrm{ID}_i}, \mathsf{ct}_{\mathrm{ID}_i}, \mathsf{td}_{\mathrm{ID}_j}, \mathsf{ct}_{\mathrm{ID}_j})$*: It takes two ciphertexts* $\mathsf{ct}_{\mathrm{ID}_i}$, $\mathsf{ct}_{\mathrm{ID}_j}$ *and two trapdoors* $\mathsf{td}_{\mathrm{ID}_i}$, $\mathsf{td}_{\mathrm{ID}_j}$ *for identities* ID_i, ID_j*, respectively, as inputs, and outputs* 0 *or* 1.

Correctness. Since IBEET is an IBE scheme, it should be guaranteed the correctness of the decryption algorithm: For any identity ID and message $M \in \mathcal{M}$,

$$\mathsf{Dec}(pp, d_{\mathrm{ID}}, \mathsf{Enc}(pp, \mathrm{ID}, M)) = M$$

should always hold where $\mathsf{Setup}(\lambda) \to (pp, msk)$ and $\mathsf{Extract}(pp, msk, \mathrm{ID}) \to d_{\mathrm{ID}}$.

For the functionality of Td and Test algorithms, we also require the following two additional conditions to be satisfied: For any identities $\mathrm{ID}_i, \mathrm{ID}_j$ and messages $M_i, M_j \in \mathcal{M}$,

1. $\Pr[\mathsf{Test}(\mathsf{td}_{\mathrm{ID}_i}, \mathsf{ct}_{\mathrm{ID}_i}, \mathsf{td}_{\mathrm{ID}_j}, \mathsf{ct}_{\mathrm{ID}_j})] = 1$ if $\mathsf{Dec}(pp, d_{\mathrm{ID}_i}, \mathsf{ct}_{\mathrm{ID}_i}) = \mathsf{Dec}(pp, d_{\mathrm{ID}_j}, \mathsf{ct}_{\mathrm{ID}_j}) \neq \perp$,
2. $\Pr[\mathsf{Test}(\mathsf{td}_{\mathrm{ID}_i}, \mathsf{ct}_{\mathrm{ID}_i}, \mathsf{td}_{\mathrm{ID}_j}, \mathsf{ct}_{\mathrm{ID}_j})]$ is negligible in the security parameter if $\mathsf{Dec}(pp, d_{\mathrm{ID}_i}, \mathsf{ct}_{\mathrm{ID}_i}) \neq \mathsf{Dec}(pp, d_{\mathrm{ID}_j}, \mathsf{ct}_{\mathrm{ID}_j})$,

where $\mathsf{Setup}(\lambda) \to (pp, msk)$, $\mathsf{Extract}(pp, msk, \mathrm{ID}_i) \to d_{\mathrm{ID}_i}$, $\mathsf{Extract}(pp, msk, \mathrm{ID}_j) \to d_{\mathrm{ID}_j}$, $\mathsf{Enc}(pp, \mathrm{ID}_i, M_i) \to \mathsf{ct}_{\mathrm{ID}_i}$, $\mathsf{Enc}(pp, \mathrm{ID}_j, M_j) \to \mathsf{ct}_{\mathrm{ID}_j}$, $\mathsf{Td}(d_{\mathrm{ID}_i}) \to \mathsf{td}_{\mathrm{ID}_i}$, and $\mathsf{Td}(d_{\mathrm{ID}_j}) \to \mathsf{td}_{\mathrm{ID}_j}$.

Security Definitions. As the same as the security definitions of our PKEET system model, we consider two types of adversaries for our IBEET system model, Type-I adversaries who have the trapdoor for the target identity and Type-II adversaries who do not have the trapdoor information. We first describe the formal security definition for IBEET constructions against Type-I adversaries.

Definition 11 (OW-ID-CCA2 against Type-I Adversaries). *An IBEET scheme is OW-ID-CCA2 secure against Type-I adversaries if for any PPT adversary $\mathcal{A}$, the success probability of $\mathcal{A}$ in the following game with the challenge $\mathcal{C}$ is negligible in the security parameter λ:*

1. **Setup:** *$\mathcal{C}$ runs $\mathsf{Setup}(\lambda) \rightarrow (pp, msk)$ and sends the public parameter pp to $\mathcal{A}$.*
2. **Phase 1:** *$\mathcal{A}$ may query the following oracles polynomially many times and adaptively and in any order.*
   - *$\mathcal{O}^{\mathsf{Ext}}$: an oracle that on input an identity ID, returns a user ID's secret key d_{ID}.*
   - *$\mathcal{O}^{\mathsf{Dec}}$: an oracle that on input an identity ID and a ciphertext ct, runs $\mathsf{Dec}(pp, d_{\mathrm{ID}}, \mathsf{ct}) \rightarrow M'$ and outputs M'.*
   - *$\mathcal{O}^{\mathsf{Td}}$: an oracle that on input an identity ID, runs $\mathsf{Td}(d_{\mathrm{ID}}) \rightarrow \mathsf{td}_{\mathrm{ID}}$ and outputs $\mathsf{td}_{\mathrm{ID}}$.*
3. **Challenge:** *$\mathcal{A}$ submits a target identity ID^*, which was never queried to the $\mathcal{O}^{\mathsf{Ext}}$ oracle in Phase 1. $\mathcal{C}$ chooses a random message M from the message space $\mathcal{M}$, runs $\mathsf{Enc}(pp, \mathrm{ID}^*, M) \rightarrow \mathsf{ct}^*_{\mathrm{ID}^*}$, and sends $\mathsf{ct}^*_{\mathrm{ID}^*}$ to $\mathcal{A}$.*
4. **Phase 2:** *For $\mathcal{A}$'s queries, $\mathcal{C}$ responds as in Phase 1. The constraints for $\mathcal{A}$'s queries are that*
   (a) the target identity ID^* cannot be queried to the secret key extraction oracle $\mathcal{O}^{\mathsf{Ext}}$;

   (b) the pair of the target identity ID^* and the challenge ciphertext $\mathsf{ct}^*_{\mathrm{ID}^*}$ cannot be queried to the decryption oracle $\mathcal{O}^{\mathsf{Dec}}$.
5. **Guess:** *$\mathcal{A}$ outputs M'.*

The adversary $\mathcal{A}$ wins in the above game if $M = M'$ and the success probability of $\mathcal{A}$ is defined to

$$\mathbf{Adv}^{\text{OW-ID-CCA2}}_{\mathcal{A},\text{IBEET}}(\lambda) := \Pr[M = M'].$$

Now, we present the definition of the IND-ID-CCA2 security of IBEET schemes against Type-II adversaries.

Definition 12 (IND-ID-CCA2 against Type-II Adversaries). *An IBEET scheme is IND-ID-CCA2 secure against Type-II adversaries if for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game with the challenge $\mathcal{C}$ is negligible in the security parameter λ:*

1. **Setup:** *This step is the same as that of the OW-ID-CCA2 security game in Definition 11.*
2. **Phase 1:** *This step is the same as that of the OW-ID-CCA2 security game in Definition 11.*
3. **Challenge:** *$\mathcal{A}$ selects a target identity ID^*, which was never queried to the $\mathcal{O}^{\mathsf{Ext}}$ and $\mathcal{O}^{\mathsf{Td}}$ oracles in Phase 1, and two messages M_0, M_1 of the same length and passes ID^*, M_0, M_1 to $\mathcal{C}$. $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$, runs $\mathsf{Enc}(pp, \mathrm{ID}^*, M_b) \rightarrow \mathsf{ct}^*_{\mathrm{ID}^*,b}$ and sends $\mathsf{ct}^*_{\mathrm{ID}^*,b}$ to $\mathcal{A}$.*

4. **Phase 2:** *For $\mathcal{A}$'s query, $\mathcal{C}$ responds as in Phase 1. The constraints for $\mathcal{A}$'s queries are that*
   (a) *the target identity* ID^* *cannot be queried to the secret key extraction oracle* $\mathcal{O}^{\mathsf{Ext}}$ *and the trapdoor extraction oracle* $\mathcal{O}^{\mathsf{Td}}$*;*
   (b) *the pair of the target identity* ID^* *and the challenge ciphertext* $\mathsf{ct}^*_{\mathrm{ID}^*}$ *cannot be queried to the decryption oracle* $\mathcal{O}^{\mathsf{Dec}}$*.*
5. **Guess:** *$\mathcal{A}$ outputs $b' \in \{0,1\}$.*

The adversary $\mathcal{A}$ wins if $b = b'$ in the above game and the advantage of $\mathcal{A}$ is defined to

$$\mathbf{Adv}^{\text{IND-ID-CCA2}}_{\mathcal{A},\text{IBEET}}(\lambda) := \left|\Pr[b = b'] - \frac{1}{2}\right|.$$

B Our IBEET Construction

In this section, we present a description of our IBEET construction using a 3-level HIBE scheme, a strongly unforgeable one-time signature scheme, and a cryptographic hash function. Then, we look into the security of the proposed IBEET scheme against Type-I and Type-II adversaries.

Throughout this section, $[\mathrm{ID}_1.\mathrm{ID}_2.\mathrm{ID}_3]$ denotes a 3-level identity where ID_i is the i-th level identity for $i = 1, 2, 3$.

B.1 Our IBEET Scheme

Description. We describe our IBEET construction below.

- $\mathsf{Setup}(\lambda)$: It take a security parameter λ as an input and performs as follows:
  1. Generate
     (a) a 3-level HIBE scheme $\mathcal{HIBE} = (\mathcal{S}etup, \mathcal{K}eyExt, \mathcal{E}nc, \mathcal{D}ec)$,
     (b) a hash function $H : \{0,1\}^* \to \mathcal{M}$ for the message space $\mathcal{M}$ of the $\mathcal{HIBE}$ scheme, and
     (c) a digital signature scheme $\mathsf{Sig} = (\mathsf{G}, \mathsf{S}, \mathsf{V})$.
  2. Run $\mathcal{S}etup(\lambda) \to (\mathsf{mpk}, \mathsf{msk})$.
  3. Output a public parameter $pp = (\mathcal{HIBE}, H, \mathsf{Sig}, \mathsf{mpk})$ and keep a master secret key $msk = \mathsf{msk}$ private.
- $\mathsf{Extract}(pp, msk, \mathrm{ID})$: On input pp, msk, and an identity ID, it runs $\mathcal{K}eyExt(msk, \mathrm{ID}) \to d_{\mathrm{ID}}$ and outputs a user ID's secret key d_{ID}.
- $\mathsf{Enc}(pp, \mathrm{ID}, M)$: It takes pp, ID, and a message M as inputs and runs
  1. $\mathsf{G}(\lambda) \to (\mathsf{vk_s}, \mathsf{sk_s})$,
  2. $\mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}.0.\mathsf{vk_s}], M) \to C_0$,

3. $\mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}.1.\mathsf{vk_s}], H(M)) \rightarrow C_1$,
4. $\mathsf{S}(\mathsf{sk_s}, [C_0 \| C_1]) \rightarrow \sigma$.

It outputs a ciphertext $\mathsf{ct} = (\mathsf{vk_s}, C_0, C_1, \sigma)$.

- $\mathsf{Dec}(pp, d_{\mathrm{ID}}, \mathsf{ct})$: On input a ciphertext ct, parse ct to $(\mathsf{vk_s}, C_0, C_1, \sigma)$. Then, it performs as follows:
  1. Run $\mathcal{K}eyExt(d_{\mathrm{ID}}, [\mathrm{ID}.i.\mathsf{vk_s}]) \rightarrow d_{[\mathrm{ID}.i.\mathsf{vk_s}]}$ for $i = 0, 1$.
  2. Decrypt C_0 and C_1 by running $\mathcal{D}ec(d_{[\mathrm{ID}.0.\mathsf{vk_s}]}, C_0) \rightarrow M'$ and $\mathcal{D}ec(d_{[\mathrm{ID}.1.\mathsf{vk_s}]}, C_1) \rightarrow h'$.
  3. If $h' = H(M')$ and $\mathsf{V}(\mathsf{vk_s}, [C_0 \| C_1], \sigma) \rightarrow 1$, then output M'. Otherwise, output $\perp$.
- $\mathsf{Td}(d_{\mathrm{ID}})$: It takes a user ID's secret key d_{ID} as an input, runs $\mathcal{K}eyExt(d_{\mathrm{ID}}, [\mathrm{ID}.1]) \rightarrow d_{[\mathrm{ID}.1]}$, and returns $\mathsf{td}_{\mathrm{ID}} = d_{[\mathrm{ID}.1]}$.
- $\mathsf{Test}(\mathsf{td}_{\mathrm{ID}_i}, \mathsf{ct}_{\mathrm{ID}_i}, \mathsf{td}_{\mathrm{ID}_j}, \mathsf{ct}_{\mathrm{ID}_j})$: It takes trapdoors $\mathsf{td}_{\mathrm{ID}_i}, \mathsf{td}_{\mathrm{ID}_j}$ and ciphertexts $\mathsf{ct}_{\mathrm{ID}_i}$, $\mathsf{ct}_{\mathrm{ID}_j}$ for users ID_i, ID_j, respectively, as inputs. For $k = i, j$,
  1. parse $\mathsf{ct}_{\mathrm{ID}_k}$ to $(\mathsf{vk}_{\mathsf{s},\mathrm{ID}_k}, C_{\mathrm{ID}_k,0}, C_{\mathrm{ID}_k,1}, \sigma_{\mathrm{ID}_k})$,
  2. run $\mathcal{K}eyExt(\mathsf{td}_{\mathrm{ID}_k}, [\mathrm{ID}_k.1.\mathsf{vk}_{\mathsf{s},\mathrm{ID}_k}]) \rightarrow d_{[\mathrm{ID}_k.1.\mathsf{vk}_{\mathsf{s},\mathrm{ID}_k}]}$,
  3. run $\mathcal{D}ec(d_{[\mathrm{ID}_k.1.\mathsf{vk}_{\mathsf{s},\mathrm{ID}_k}]}, C_{\mathrm{ID}_k,1}) \rightarrow h'_k$.

  If $h'_i = h'_j$, then output 1. Otherwise, output 0.

Correctness. The following theorem shows the correctness of our IBEET construction.

Theorem 4. *Our IBEET construction is correct if the underlying HIBE scheme $\mathcal{HIBE}$ and signature scheme* Sig *are correct, and the employed hash function H is collision resistant.*

Proof. Let $\mathsf{ct} = (\mathsf{vk_s}, C_0, C_1, \sigma)$ be a valid ciphertext obtained by running $\mathsf{Enc}(pp, \mathrm{ID}, M)$ for any identity ID and a message M, where pp is a public parameter obtained by running $\mathsf{Setup}(\lambda)$. That is, $\mathsf{G}(\lambda) \rightarrow (\mathsf{vk_s}, \mathsf{sk_s})$, $\mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}.0.\mathsf{vk_s}], M) \rightarrow C_0$, $\mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}.1.\mathsf{vk_s}], H(M)) \rightarrow C_1$, and $\mathsf{S}(\mathsf{sk_s}, [C_0 \| C_1]) \rightarrow \sigma$. Since $\mathcal{HIBE}$ is correct,

$$\begin{aligned}\mathcal{D}ec(d_{[\mathrm{ID}.0.\mathsf{vk_s}]}, C_0) &= \mathcal{D}ec(d_{[\mathrm{ID}.0.\mathsf{vk_s}]}, \mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}.0.\mathsf{vk_s}], M)) \\ &\rightarrow M' = M\end{aligned}$$

and

$$\begin{aligned}\mathcal{D}ec(d_{[\mathrm{ID}.1.\mathsf{vk_s}]}, C_1) &= \mathcal{D}ec(d_{[\mathrm{ID}.1.\mathsf{vk_s}]}, \mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}.1.\mathsf{vk_s}], H(M))) \\ &\rightarrow h' = H(M)\end{aligned}$$

where $d_{[\mathrm{ID}.i.\mathsf{vk_s}]}$ is an outcome of $\mathcal{K}eyExt(d_{\mathrm{ID}}, [\mathrm{ID}.i.\mathsf{vk_s}])$ for $i = 0, 1$. Hence, $h' = H(M')$ always holds. Further, since Sig is correct,

$$\mathsf{V}(\mathsf{vk_s}, [C_0\|C_1], \sigma) = \mathsf{V}(\mathsf{vk_s}, [C_0\|C_1], \mathsf{S}(\mathsf{sk_s}, [C_0\|C_1])) \to 1.$$

Therefore, $\mathsf{Dec}(pp, d_{\mathrm{ID}}, \mathsf{ct})$ always outputs the correct message M.

Let $\mathsf{ct}_{\mathrm{ID}_k} = (\mathsf{vk}_{\mathsf{s},\mathrm{ID}_k}, C_{\mathrm{ID}_k,0}, C_{\mathrm{ID}_k,1}, \sigma_{\mathrm{ID}_k})$ be a valid ciphertext, obtained by running $\mathsf{Enc}(pp, \mathrm{ID}_k, M_k)$ for any identity ID_k and message M_k with $k = i, j$, where pp is the public parameter generated by running $\mathsf{Setup}(\lambda)$. Since $\mathcal{HIBE}$ is correct,

$$\mathcal{D}ec(d_{[\mathrm{ID}_k.1.\mathsf{vk}_{\mathsf{s},\mathrm{ID}_k}]}, C_{\mathrm{ID}_k,1}) \to h'_k = H(M_k)$$

where $d_{[\mathrm{ID}_k.1.\mathsf{vk}_{\mathsf{s},\mathrm{ID}_k}]}$ is an outcome of $\mathcal{K}eyExt(d_{\mathrm{ID}_k}, [\mathrm{ID}_k.1.\mathsf{vk}_{\mathsf{s},\mathrm{ID}_k}])$ for $k = i, j$. Hence, if $M_i = M_j$, then $h'_i = H(M_i) = H(M_j) = h'_j$ and so the Test algorithm always outputs 1. On the other hand, if $\mathsf{Dec}(pp, \mathsf{sk}_i, \mathsf{ct}_i) \neq \mathsf{Dec}(pp, \mathsf{sk}_j, \mathsf{ct}_j)$, then $M_i \neq M_j$ and so $h'_i = H(M_i) \neq H(M_j) = h'_j$ with overwhelming probability because H is collision resistant. Therefore, the Test algorithm outputs 1 with negligible probability.

From the above, it is proved that our IBEET construction is correct. □

B.2 Security Analysis

Now, we investigate the security of our IBEET constructions against Type-I and Type-II adversaries. The following theorem demonstrates that our IBEET construction is IND-ID-CCA2 secure against Type-II adversaries who do not have a trapdoor for an equality test on the challenge ciphertext.

Theorem 5 (IND-ID-CCA2). *If $\mathcal{HIBE}$ is an IND-ID-CPA secure 3-level HIBE scheme and* Sig *is a strongly unforgeable one-time signature scheme, then the proposed IBEET scheme exploiting $\mathcal{HIBE}$ and* Sig *is IND-ID-CCA2 secure against Type-II adversaries in the standard model.*

More precisely, if there is no PPT adversary that breaks the IND-ID-CPA security of $\mathcal{HIBE}$ with at least $\varepsilon_{\mathcal{HIBE}}$ advantage and there is no PPT adversary that breaks the strong unforgeability of Sig *with at least $\varepsilon_{\mathsf{Sig}}$ success probability, then for any PPT adversary that breaks the IND-ID-CCA2 security of the proposed IBEET scheme, its advantage is bounded above by $2\varepsilon_{\mathcal{HIBE}} + \dfrac{3\varepsilon_{\mathsf{Sig}}}{2}$.*

Proof. The proof of this theorem is very similar to that of Theorem 2, which shows the IND-CCA2 security of our PKEET construction. Similarly to the proof of Theorem 2, we begin with defining the following three games.

Game$_0$: This game is the same as the original IND-ID-CCA2 security game in Definition 12. Denote the challenge ciphertext for the target identity ID^* by $\mathsf{ct}^*_{\mathrm{ID}^*} = (\mathsf{vk}^*_{\mathsf{s},\mathrm{ID}^*}, C^*_{\mathrm{ID}^*,0}, C^*_{\mathrm{ID}^*,1}, \sigma^*_{\mathrm{ID}^*})$.

Game$_1$: This game is almost the same as **Game**$_0$, except that if $\mathcal{A}$ queries the oracle $\mathcal{O}^{\mathsf{Dec}}(\mathrm{ID}^*, \cdot)$ on $\mathsf{ct}_{\mathrm{ID}^*} = (\mathsf{vk}_{\mathsf{s},\mathrm{ID}^*}, C_{\mathrm{ID}^*,0}, C_{\mathrm{ID}^*,1}, \sigma_{\mathrm{ID}^*})$ such that $\mathsf{vk}_{\mathsf{s},\mathrm{ID}^*} =$

$\mathsf{vk_s}^*_{,\mathrm{ID}^*}$, $\mathsf{ct}_{\mathrm{ID}^*} \neq \mathsf{ct}^*_{\mathrm{ID}^*}$ and $\mathsf{V}(\mathsf{vk}_{\mathsf{s},\mathrm{ID}^*}, [C_{\mathrm{ID}^*,0} \| C_{\mathrm{ID}^*,1}], \sigma_{\mathrm{ID}^*}) \to 1$, then the challenger $\mathcal{C}$ stops the interaction and sets $\mathcal{A}$'s answer at random.

Game$_2$: This game is almost the same as **Game**$_1$, except for the challenger's response in the challenge phase. In **Game**$_0$ and **Game**$_1$, the original ciphertext $\mathsf{ct}^*_{\mathrm{ID}^*}$ has the form $(\mathsf{vk_s}^*_{,\mathrm{ID}^*}, C^*_{\mathrm{ID}^*,0}, C^*_{\mathrm{ID}^*,1}, \sigma^*_{\mathrm{ID}^*})$ where $\mathsf{G}(\lambda) \to (\mathsf{vk_s}^*_{,\mathrm{ID}^*}, \mathsf{sk_s}^*_{,\mathrm{ID}^*})$, $\mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}^*.0.\mathsf{vk_s}^*_{,\mathrm{ID}^*}], M_b) \to C^*_{\mathrm{ID}^*,0}$, $\mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}^*.1.\mathsf{vk_s}^*_{,\mathrm{ID}^*}], H(M_b)) \to C^*_{\mathrm{ID}^*,1}$, and $\mathsf{S}(\mathsf{sk_s}^*_{\mathrm{ID}^*}, [C^*_{\mathrm{ID}^*,0} \| C^*_{\mathrm{ID}^*,1}]) \to \sigma^*_{\mathrm{ID}^*}$, and b is a random bit chosen by $\mathcal{C}$. On the other hand, at the beginning of the challenge phase in **Game**$_2$, $\mathcal{A}$ issues a target identity ID^* and two messages M_0, M_1, and sends $\mathcal{C}$ them. Then, $\mathcal{C}$ tosses two unbiased coins a and b. If $a = 0$, $\mathcal{C}$ generates the challenge ciphertext by using M_b and $H(M_{1-b})$, instead of M_b and $H(M_b)$, respectively. Otherwise, $\mathcal{C}$ uses M_{1-b} and $H(M_b)$ for generating C_0 and C_1, respectively.

Let ε_i be the advantage of $\mathcal{A}$ in **Game**$_i$ for $i = 1, 2, 3$. Then, we obtain the following relations among ε_i's:

(i) $\varepsilon_0 - \varepsilon_1 < \dfrac{3\varepsilon_{\mathsf{Sig}}}{2}$,

(ii) $\varepsilon_1 - \varepsilon_2 < 2\varepsilon_{\mathcal{HIBE}}$,

(iii) $\varepsilon_2 = 0$.

We note that the relations (i) and (iii) can be obtained from slight modifications of proofs of Lemmas 1 and 3, respectively, adapted to the identity-based setting. However, for the relation (ii), we need a more careful modification from the proof of Lemma 2: While our PKEET scheme requires the IND-sID-CPA security of $\mathcal{HIBE}$, our IBEET scheme requires the IND-ID-CPA security. Hence, by considering this difference, the simulator $\mathcal{S}_{\mathcal{HIBE}}$ in the proof of Lemma 2 should be adjusted.

The following is the full description of $\mathcal{S}_{\mathcal{HIBE}}$'s behaviour for the security analysis of our IBEET construction. We construct a simulator $\mathcal{S}_{\mathcal{HIBE}}$ that breaks the IND-ID-CPA security of $\mathcal{HIBE}$ by using $\mathcal{A}$. Denote the challenger of the IND-ID-CPA security game for $\mathcal{HIBE}$ by $\mathcal{C}_{\mathcal{HIBE}}$. Once $\mathcal{S}_{\mathcal{HIBE}}$ receives the master public key mpk of $\mathcal{HIBE}$ from $\mathcal{C}_{\mathcal{HIBE}}$, $\mathcal{S}_{\mathcal{HIBE}}$ generates a hash function H and a one-time signature Sig, and sends a public parameter $pp = (\mathcal{HIBE}, H, \mathsf{Sig}, \mathsf{mpk})$ to $\mathcal{A}$. As for $\mathcal{A}$'s queries, $\mathcal{S}_{\mathcal{HIBE}}$ can respond correctly by exploiting her own oracles provided by $\mathcal{C}_{\mathcal{HIBE}}$.

In the challenge phase, $\mathcal{A}$ chooses a target identity ID^* and two messages M_0, M_1, and sends $\mathcal{S}_{\mathcal{HIBE}}$ them. $\mathcal{S}_{\mathcal{HIBE}}$ first runs $\mathsf{G}(\lambda) \to (\mathsf{vk_s}^*_{,\mathrm{ID}^*}, \mathsf{sk_s}^*_{,\mathrm{ID}^*})$ and tosses a coin $\alpha \in \{0, 1\}$. Depending on the value α, the challenge ciphertext for $\mathcal{A}$ is differently computed: If $\alpha = 0$, $\mathcal{S}_{\mathcal{HIBE}}$ sends $\mathcal{C}_{\mathcal{HIBE}}$ a (3-level) target identity $[\mathrm{ID}^*.\alpha.\mathsf{vk_s}^*]$ along with two messages M_0 and M_1. Once $\mathcal{S}_{\mathcal{HIBE}}$ receives the challenge ciphertext $C^*_{\mathrm{ID}^*,0,b}$ for a random bit b chosen by $\mathcal{C}_{\mathcal{HIBE}}$, $\mathcal{S}_{\mathcal{HIBE}}$ selects a random bit $\beta \in \{0, 1\}$ and sends $\mathcal{A}$

$$\mathsf{ct}^*_{\mathrm{ID}^*} = (\mathsf{vk_s}^*_{,\mathrm{ID}^*}, C^*_{\mathrm{ID}^*,0,b}, C^*_{\mathrm{ID}^*,1,\beta} = \mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}^*.1.\mathsf{vk}_{\mathsf{s},\mathrm{ID}^*}], H(M_\beta)), \sigma^*_{\mathrm{ID}^*})$$

where $\mathsf{S}(\mathsf{sk_s}^*_{,\mathrm{ID}^*}, [C^*_{\mathrm{ID}^*,0,b} \| C^*_{\mathrm{ID}^*,1,\beta}]) \rightarrow \sigma^*_{\mathrm{ID}^*}$. On the other hand, if $\alpha = 1$, $\mathcal{S}_{\mathcal{HIBE}}$ sends $\mathcal{C}_{\mathcal{HIBE}}$ a (3-level) target identity $[\mathrm{ID}^*.\alpha.\mathsf{vk_s}^*]$ along with two messages $H(M_0)$ and $H(M_1)$, not M_0 and M_1, respectively. Once $\mathcal{S}_{\mathcal{HIBE}}$ receives the challenge ciphertext $C^*_{\mathrm{ID}^*,1,b}$ for a random bit b chosen by $\mathcal{C}_{\mathcal{HIBE}}$, $\mathcal{S}_{\mathcal{HIBE}}$ selects a random bit $\beta \in \{0,1\}$ and sends $\mathcal{A}$

$$\mathsf{ct}^*_{\mathrm{ID}^*} = (\mathsf{vk_s}^*_{,\mathrm{ID}^*}, C^*_{\mathrm{ID}^*,0,\beta} = \mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}^*.0.\mathsf{vk}_{\mathsf{s},\mathrm{ID}^*}], M_\beta), C^*_{\mathrm{ID}^*,1,b}, \sigma^*_{\mathrm{ID}^*})$$

where $\mathsf{S}(\mathsf{sk_s}^*_{,\mathrm{ID}^*}, [C^*_{\mathrm{ID}^*,0,\beta} \| C^*_{\mathrm{ID}^*,1,b}]) \rightarrow \sigma^*_{\mathrm{ID}^*}$.

After the challenge phase, $\mathcal{S}_{\mathcal{HIBE}}$ responds to $\mathcal{A}$'s all queries as before the challenge phase. The difference between before and after the challenge phase is that if $\mathcal{A}$ issues a decryption query on a valid ciphertext using $\mathsf{vk_s}^*_{,\mathrm{ID}^*}$, then $\mathcal{S}_{\mathcal{HIBE}}$ stops the interaction with $\mathcal{A}$ and sets $\mathcal{A}$'s output to be a random coin. We note that $\mathcal{A}$ cannot issue a key extraction query $\mathcal{O}^{\mathsf{Ext}}$ on the challenge identity ID*, a decryption query $\mathcal{O}^{\mathsf{Dec}}$ on the pair of the challenge identity and ciphertext $(\mathrm{ID}^*, \mathsf{ct}^*_{\mathrm{ID}^*})$, and a trapdoor extraction query $\mathcal{O}^{\mathsf{Td}}$ on the challenge identity ID*. For all other queries, $\mathcal{S}_{\mathcal{HIBE}}$ can respond correctly by accessing her own oracles offered by $\mathcal{C}_{\mathcal{HIBE}}$.

Finally, once $\mathcal{A}$ outputs an answer b', $\mathcal{S}_{\mathcal{HIBE}}$ forwards b' to $\mathcal{C}_{\mathcal{HIBE}}$. We omit the analysis of the advantage of $\mathcal{S}_{\mathcal{HIBE}}$, but it is the same as that in the proof of Lemma 2 if the notation $\mathbf{Adv}^{\text{IND-sID-CPA}}_{\mathcal{S}_{\mathcal{HIBE}},\mathcal{HIBE}}(\lambda)$ is replaced with $\mathbf{Adv}^{\text{IND-ID-CPA}}_{\mathcal{S}_{\mathcal{HIBE}},\mathcal{HIBE}}(\lambda)$.

Overall, from (i), (ii), and (iii), we have

$$\varepsilon_0 \leq 2\varepsilon_{\mathcal{HIBE}} + \frac{3\varepsilon_{\mathsf{Sig}}}{2}.$$

□

Next, we show that our IBEET construction is OW-ID-CCA2 secure against Type-I adversaries who have a trapdoor for an equality test on the challenge ciphertext.

Theorem 6 (OW-ID-CCA2). *If $\mathcal{HIBE}$ is an IND-ID-CPA secure 3-level HIBE scheme, H is a one-way hash function, and* Sig *is a strongly unforgeable one-time signature scheme, then the proposed IBEET scheme exploiting $\mathcal{HIBE}$, H, and* Sig*, is OW-ID-CCA2 secure against Type-I adversaries in the standard model.*

More precisely, if there is no PPT adversary that breaks one of the IND-ID-CPA security of $\mathcal{HIBE}$, the one-wayness of H, the strong unforgeability of Sig *with at least $\varepsilon_{\mathcal{HIBE}}$ advantage, ε_H success probability, and $\varepsilon_{\mathsf{Sig}}$ success probability, respectively, then for any PPT adversary that breaks the OW-ID-CCA2 security of the proposed IBEET construction, its success probability is bounded above by $4\varepsilon_{\mathcal{HIBE}} + \varepsilon_H + \varepsilon_{\mathsf{Sig}}$.*

Proof. The proof of this theorem is very similar to that of Theorem 3, which shows the OW-CCA2 security of our PKEET construction. Similarly to the proof of Theorem 3, we begin by defining the following two games.

Game$_\mathbf{0}$: This is the same as the original OW-ID-CCA2 security game in Definition 11. Denote the challenge ciphertext for the target identity ID* by $\mathsf{ct}^*_{\mathrm{ID}^*} = (\mathsf{vk_s}^*_{,\mathrm{ID}^*}, C^*_{\mathrm{ID}^*,0}, C^*_{\mathrm{ID}^*,1}, \sigma^*_{\mathrm{ID}^*})$.

Game$_\mathbf{1}$: This game is almost the same as **Game**$_0$, except that if $\mathcal{A}$ queries the oracle $\mathcal{O}^{\mathsf{Dec}}(\mathrm{ID}^*, \cdot)$ on $\mathsf{ct}_{\mathrm{ID}^*} = (\mathsf{vk}_{\mathsf{s},\mathrm{ID}^*}, C_{\mathrm{ID}^*,0}, C_{\mathrm{ID}^*,1}, \sigma_{\mathrm{ID}^*})$ such that $\mathsf{vk}_{\mathsf{s},\mathrm{ID}^*} = \mathsf{vk_s}^*_{,\mathrm{ID}^*}$, $\mathsf{ct}_{\mathrm{ID}^*} \neq \mathsf{ct}^*_{\mathrm{ID}^*}$ and $\mathsf{V}(\mathsf{vk}_{\mathsf{s},\mathrm{ID}^*}, [C_{\mathrm{ID}^*,0} \| C_{\mathrm{ID}^*,1}], \sigma_{\mathrm{ID}^*}) \to 1$, then the challenger $\mathcal{C}$ stops the interaction and sets $\mathcal{A}$'s answer at random.

Let ε_i be the success probability of $\mathcal{A}$ in **Game**$_i$ for $i = 0, 1$. First, using a similar argument in the proof of Theorem 3, we obtain the relation

$$\varepsilon_0 - \varepsilon_1 < \varepsilon_{\mathsf{Sig}}. \tag{6}$$

Second, as in the proof of Theorem 3, we can define the simulation $\mathcal{PS}$, which is exactly the same as a normal challenger in **Game**$_1$, except that the simulator chooses two different messages M_0', M_1' at random and generates two ciphertexts

$$C^*_{\mathrm{ID}^*,0} = \mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}^*.0.\mathsf{vk_s}^*_{,\mathrm{ID}^*}], M_0') \text{ and}$$
$$C^*_{\mathrm{ID}^*,1} = \mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}^*.1.\mathsf{vk_s}^*_{,\mathrm{ID}^*}], H(M_1'))$$

where ID* is the challenge identity and $(\mathsf{vk_s}^*_{,\mathrm{ID}^*}, \mathsf{sk_s}^*_{,\mathrm{ID}^*})$ is an outcome of the algorithm $\mathsf{S}(\lambda)$. Then, we have the relation

$$\Pr[\mathcal{A} \to M_1' \text{ in } \mathcal{PS}] < \varepsilon_H \tag{7}$$

as in the proof of Theorem 3.

Finally, we can evaluate the success probability of the adversary $\mathcal{A}$ in **Game**$_1$ by constructing a simulator $\mathcal{S}_{\mathcal{HIBE}}$ that breaks the IND-ID-CPA security of $\mathcal{HIBE}$ using $\mathcal{A}$. $\mathcal{S}_{\mathcal{HIBE}}$ is very similar to that in the proof of Theorem 3, but we need a careful modification by considering that the security requirement of $\mathcal{HIBE}$ for our IBEET construction is the IND-ID-CPA security, not the IND-sID-CPA security, as in the proof of Theorem 5. We provide the full description of $\mathcal{S}_{\mathcal{HIBE}}$'s behaviour below.

Denote the challenger of the IND-ID-CPA security game for $\mathcal{HIBE}$ by $\mathcal{C}_{\mathcal{HIBE}}$. Once $\mathcal{S}_{\mathcal{HIBE}}$ receives the master public key mpk of $\mathcal{HIBE}$ from $\mathcal{C}_{\mathcal{HIBE}}$, $\mathcal{S}_{\mathcal{HIBE}}$ generates a hash function H and a one-time signature Sig, and sends a public parameter $pp = (\mathcal{HIBE}, H, \mathsf{Sig}, \mathsf{mpk})$ to $\mathcal{A}$. As for $\mathcal{A}$'s queries, $\mathcal{S}_{\mathcal{HIBE}}$ can respond correctly by using her own oracles offered by $\mathcal{C}_{\mathcal{HIBE}}$.

In the challenge phase, $\mathcal{A}$ sends a target identity ID* to $\mathcal{S}_{\mathcal{HIBE}}$. Then, $\mathcal{S}_{\mathcal{HIBE}}$ selects two messages M_0 and M_1 at random, runs $\mathsf{G}(\lambda) \to (\mathsf{vk_s}^*_{,\mathrm{ID}^*}, \mathsf{sk_s}^*_{,\mathrm{ID}^*})$, and sends $\mathcal{C}_{\mathcal{HIBE}}$ a 3-level target identity $[\mathrm{ID}^*.0.\mathsf{vk_s}^*_{,\mathrm{ID}^*}]$ along with two messages M_0, M_1. Once $\mathcal{S}_{\mathcal{HIBE}}$ receives $C^*_{\mathrm{ID}^*,0,b}$ that is a ciphertext of the message M_b with respect to the target identity $[\mathrm{ID}^*.0.\mathsf{vk_s}^*_{,\mathrm{ID}^*}]$, where b is a random bit chosen by $\mathcal{C}_{\mathcal{HIBE}}$, $\mathcal{S}_{\mathcal{HIBE}}$ selects a random bit $\beta \in \{0, 1\}$, and sends

$$\mathsf{ct}^*_{\mathrm{ID}^*} = (\mathsf{vk_s}^*_{,\mathrm{ID}^*}, C^*_{\mathrm{ID}^*,0,b}, C^*_{\mathrm{ID}^*,1,\beta}, \sigma^*_{\mathrm{ID}^*})$$

to $\mathcal{A}$ where $C^*_{\mathrm{ID}^*,1,\beta} = \mathcal{E}nc(\mathsf{mpk}, [\mathrm{ID}^*.1.\mathsf{vk}^*_{\mathsf{s},\mathrm{ID}^*}], H(M_\beta))$ and $\mathsf{S}(\mathsf{sk}^*_{\mathsf{s},\mathrm{ID}^*}, [C^*_{\mathrm{ID}^*,0,b} \| C^*_{\mathrm{ID}^*,1,\beta}]) \to \sigma^*_{\mathrm{ID}^*}$.

After the challenge phase, $\mathcal{S}_{\mathcal{HIBE}}$ responds to $\mathcal{A}$'s all queries as before the challenge phase. The difference between before and after the challenge phase is that if $\mathcal{A}$ issues a decryption query on a valid ciphertext using $\mathsf{vk}^*_{\mathsf{s},\mathrm{ID}^*}$, then $\mathcal{S}_{\mathcal{HIBE}}$ stops the interaction with $\mathcal{A}$ and sets $\mathcal{A}$'s output to be a random coin. We note that $\mathcal{A}$ cannot issue a key extraction query $\mathcal{O}^{\mathsf{Ext}}$ on the target identity ID^*, a decryption query $\mathcal{O}^{\mathsf{Dec}}$ on the pair of the challenge identity and ciphertext $(\mathrm{ID}^*, \mathsf{ct}^*_{\mathrm{ID}^*})$. For all other queries, $\mathcal{S}_{\mathcal{HIBE}}$ can respond correctly by accessing her own oracles offered by $\mathcal{C}_{\mathcal{HIBE}}$.

Finally, $\mathcal{A}$ outputs either a message M' or $\perp$ to $\mathcal{S}_{\mathcal{HIBE}}$. If $M' = M_\beta$, then $\mathcal{S}_{\mathcal{HIBE}}$ returns $b' = \beta$ to $\mathcal{C}_{\mathcal{HIBE}}$. Otherwise, $\mathcal{S}_{\mathcal{HIBE}}$ sets a bit b at random and returns it to $\mathcal{C}_{\mathcal{HIBE}}$. The analysis of the advantage of $\mathcal{S}_{\mathcal{HIBE}}$ is exactly the same as in the proof of Theorem 3 and so we have

$$\frac{1}{4}(\varepsilon_1 - \varepsilon_H) < \varepsilon_{\mathcal{HIBE}}. \tag{8}$$

Therefore, from Equations (6)-(8), we obtain $\varepsilon_0 < 4\varepsilon_{\mathcal{HIBE}} + \varepsilon_H + \varepsilon_{\mathsf{Sig}}$. □